Oracle Database 11gR2 Upgrade Companion (Version 2.80)

8/16/2012

Welcome to the Oracle Database 11gR2 Upgrade Companion. This Upgrade Companion helps you to upgrade from either Oracle9i Release 2 (9.2) or Oracle Database 10g to Oracle Database 11g Release 2, and includes pre-upgrade, upgrade, and post-upgrade guidance. Oracle continually updates this document as new information becomes available. Please check this document prior to performing any upgrade.

NOTE: The Upgrade Companion is an instructional document that serves as a companion to the Oracle Database documentation set. The Upgrade Companion:

- Does not supply automation tools
- Does not replace the Oracle Database 11g Upgrade Guide
- Describes the upgrade for generic database only. Database upgrade requirements for customer or Oracle applications should be factored into the upgrade as recommended by the product documentation.

For advice or onsite assistance during a database upgrade, see the Accelerate Technology Adoption page or the Oracle Consulting Upgrade Services page. Oracle Advanced Customer Services helps you make better IT decisions by providing you with the option to develop a personalized technology strategy and long-term operational plan for a successful transition to new Oracle capabilities. Oracle Consulting is a low risk cost-effective choice for Oracle upgrades. The Oracle Consulting service can be provided in partnership with your in-house staff, in close coordination with your chosen service provider, or as a remote service.

For application upgrades, see your application documentation and My Oracle Support. For your convenience, the following list shows some common E-Business Suite My Oracle Support documents. For a complete list of documents, see My Oracle Support.

Oracle E-Business Suite:

- Document 881505.1: Interoperability Notes Oracle EBS 11i with Oracle Database 11gR2 (11.2.0)
- Document 1058763.1: Interoperability Notes EBS R12 with Database 11gR2

Modifications

Version 2.80 August 16, 2012

- Modified guide to include changes introduced with Oracle Release 11.2.0.3.

Version 2.70 August 10, 2011

Behavior Changes

- Added content for JOB_QUEUE_PROCESSES change.
- Added content for Datafile Write Errors change.

Version 2.60 June 28, 2011

Modified guide to include changes introduced with Oracle Release 11.2.0.2.

Version 2.50 March 29, 2011

Upgrade Planning > Technical Planning

- Added references to the Upgrade/Patch Planner and Certify Toolset under "Scripts and Tools"

Version 2.40 August 2, 2010

Modified reference links to point to My Oracle Support instead of MetaLink.

Some documents were still pointing to the beta document site. Modified these links.

Version 2.30 September 30, 2009

Published Guide to all customers now that 11gR2 has been released.

Removed disclaimer for beta documents. 11gR2 is no longer a beta release.

Documentation reference links point to 11gR2 production documentation.

Version 2.20 August 21, 2009

Behavior Changes

- Added content for TIMESTAMP WITH TIME ZONE data.

Version 2.10 May 11, 2009

Added disclaimer for beta documents.

Modified the documentation reference links to point to 11gR2 beta documentation.

Version 2.00 April 30, 2009

Version 2.00 is the first release of the Oracle Database 11gR2 Upgrade Companion.

Contents

Feedback

To help us improve this guide or to notify us of any issues that you have encountered with the guide, send your comments and suggestions to Vickie.Carbonneau@oracle.com, Technical Advisor, Center of Expertise (CoE). We look forward to your feedback.

Introduction

The best practices presented in this section are derived from the knowledge of Oracle technical staff and offer 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 database configuration, and testing
- Prepare and Preserve: Information related to preserving and preparing the source environment for the Oracle Database 11g 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 11g

NOTE: Be sure to validate and adjust the upgrade steps repeatedly in your test environment. Your final upgrade plan and execution steps should run smoothly during testing before you perform the upgrade in your production environment.

The following sample workflow illustrates a test and validate approach:

1. Upgrade Planning - Evaluate and document the plan for configuring and testing the upgrade procedure in your test environment
    - The documented plan resulting for this step will be relevant for Test, Stage, and Production environments
2. Prepare and Preserve - 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 your production database.
4. Post-upgrade - Use the tips and techniques documented here to ensure your test environment is performing up to a standard required for production
5. At this point, you have upgraded the test environment. Consider the following:
    - Have you adjusted your plan to include everything you learned from the test upgrade?
        - During your production upgrade, an accurate plan is important to avoid problems that were encountered during the test upgrade
    - Are you comfortable that you have a repeatable plan to upgrade production?
        - If not, test the upgrade procedure again

- Are you comfortable that the system was tested adequately for functionality and stability and will adhere to all of your performance and availability requirements?
  - Sufficient and proper testing is critical to avoid problems after upgrading to Oracle Database 11g
- Have you tested your fallback plans and procedures?

6. Once you are comfortable that you can move on to upgrade the Stage or Production environment, execute steps 2 through 4 on that environment.

Executing a well-defined plan helps to mitigate risk and provides clear backup and 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 more time that you spend planning and testing the upgrade, the more successful the upgrade will be.

Starting with the first patch set for Oracle Database 11g Release 2 (11.2.0.2), Oracle Database patch sets are full installations of the Oracle Database software. Beginning with the release 11.2.0.2 patch set, Oracle recommends that you perform an out-of-place patch set upgrade because it requires much less downtime and is safer as it does not require patching an ORACLE_HOME that is already being used in production. For an out-of-place patch set upgrade, install the patch set into a new, separate Oracle home location. After you install the patch upgrade, you then migrate the Oracle Database from the older Oracle home. See My Oracle Support Document 1189783.1: Important Changes to Oracle Database Patch Sets Starting With 11.2.0.2 for more details.

Documentation Roadmap and Planning

The information provided below is intended to supplement the *Oracle Database Upgrade Guide*. This section provides a roadmap to Oracle documentation that you should review when defining your upgrade plan.

- Review the detailed instructions provided in the Oracle Database 11g Upgrade Guide.
- Carefully review the Oracle Database Installation Guide for your specific Operating System; this guide describes how to install and configure Oracle Database 11g database.
- Review the new features for Oracle Database 11g Release 2 described in Oracle Database 11g New Features Guide
- Review My Oracle Support Document 884232.1: 11gR2 Install (Non-RAC): Understanding New Changes With All New 11.2 Installer which describes 11.2 changes with the Oracle Universal Installer (OUI).
- Review the Oracle Database Administrator's Guide 11g.
- Review the Oracle Database Administrator Reference 11g for Linux and UNIX Based Operating Systems if you're upgrading in a UNIX/Linux environment.
- Review My Oracle Support Document 264.1:Upgrade Advisor: Database from 9.2 to 11.2 for other upgrade assistance. The Upgrade Advisor provides a guided path to plan for and execute an upgrade of an Oracle Database from Oracle Database 9i Release 2 (9.2) to Oracle Database 11g Release 2 (11.2).
- Review My Oracle Support Document 251.1:Upgrade Advisor: Database from 10.2 to 11.2 for other upgrade assistance. The Upgrade Advisor provides a guided path to plan for and execute an upgrade of an Oracle Database 10g release 2 (10.2) to Oracle Database 11g release 2 (11.2).
- If upgrading RAC/Oracle Clusterware databases, review My Oracle Support Document 810394.1: RAC Assurance Support Team: RAC and Oracle Clusterware Starter Kit and Best Practices
- As you perform each phase, ensure the steps are well defined, carefully documented, and note the amount of time required to complete each phase.

Technical Planning

NOTE: For best results, only make database changes which are related to the database upgrade.

- Determine the Upgrade Path
- Choose an Upgrade Method - Oracle Database 11g 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 Single Instance databases and Oracle RAC
      - 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
    - Disadvantage
      - Offers less control over individual upgrade steps
  - Manual Upgrade
    - Command-line upgrade using Oracle supplied SQL scripts and utilities (My Oracle Support Document 837570.1: Complete Checklist for Manual Upgrades to 11gR2 provides a checklist for manual upgrades.)
    - Advantage
      - The DBA controls every step of the upgrade process
    - Disadvantages
      - More work
      - Manual checks required of spool logs for errors
      - More error prone
      - More difficult to automate
  - Export/Import (see also Chapter 7 of the *Oracle Database Upgrade Guide*)
    - Full or Partial Export followed by full or partial import into Oracle Database 11g
    - Advantages
      - Defragments the data
      - Restructures the database
      - Enables the copying of specified database objects or users
      - Serves as a backup archive
    - Disadvantage
      - Can take a long time
      - Introduces other factors that need to be considered when trying to understand performance profile differences detected after the upgrade (e.g. the change in physical location of data relative to other data, forgetting to recreate an index after import, etc.)
- Document the steps to reproduce the environment and make sure to note any changes in the configuration.
- Ensure that all elements of your environment are certified to work together

Notes

- Document 369644.1: Answers To FAQ For Restoring Or Duplicating Between Different Versions And Platforms
- Document 837570.1: Complete Checklist for Manual Upgrades to 11gR2
- Document 161818.1: Oracle Server (RDBMS) Releases Support Status Summary
- Document 169706.1: Oracle Database on AIX, HP-UX,Linux, Mac OS X, Solaris,Tru64 Unix Operating Systems Installation and Configuration Requirements Quick Reference (8.0.5 to 11.1)
- Document 880782.1: ALERT: Oracle 11g Release 2 (11.2) Support Status and Alerts
- Document 1272288.1: 11.2.0.2.X Grid Infrastructure Bundle/PSU Known Issues
- Document 1152016.1: Master Note For Oracle Database Upgrades and Migrations
- Document 1392633.1: Things to Consider Before Upgrading to 11.2.0.3 to Avoid Poor Performance or Wrong Results

HA Documentation

- Oracle Database High Availability Overview

Scripts and Tools

- Document 847410.5: My Oracle Support Help - Patches and Updates: Patch and Upgrade Plans
- Document 1295603.1: Locate Database Server Certification Information on My Oracle Support

    - If Oracle Enterprise Manager Grid Control (EMGC) is installed, use My Oracle Support Document 412431.1: Oracle Enterprise Manager 10g Grid Control Certification Checker to ensure that EMGC is certified with all Oracle & non-Oracle targets prior to the upgrade.
    - If using ASM review the Oracle Database Storage Administrator's Guide: Administering ASM Instances for information about operating with different releases of ASM and Database Instances Simultaneously. Also check My Oracle Support Document 337737.1: Oracle Clusterware - ASM - Database Version Compatibility for ASM, Oracle Clusterware and Database Instance certification.
    - My Oracle Support certification: Always ensure that your desired OS and Oracle combination is certified. Go to My Oracle Support, click on the "Certifications" tab, and follow the steps to verify your OS and Oracle combination in the CERTIFY section.
    - My Oracle Support certification: Always ensure that your desired Product combinations are certified to work together. Go to My Oracle Support, click on the "Certifications" tab, and follow the steps to verify your Product combinations in the CERTIFY section.
- Determine if there are any known issues, Alerts, OS Patches, Database Patches, Patch Set Updates (PSUs), or Critical Patches Updates (CPUs) available. Ensure that all the critical patchsets, PSUs, and CPUs are applied.
    - Always apply all necessary patches in the target `$ORACLE_HOME` before doing the upgrade. This will omit an additional recompilation process and save time.
    - Check My Oracle Support Document 161818.1: Oracle Database (RDBMS) Releases Support Status Summary - click on your target release (RELEASE column) in the colored table - click on "Availability and Known issues" or "Known issues and alerts" for your desired target patchset - check the ALERTS section as well as the UPGRADE ISSUEs section for any known problems. Apply the recommended patches to your `$ORACLE_HOME`.
    - Review "Behavior Changes" tabs in this document.
    - Review the "Patches Recommended" section in this document for patches to apply before the upgrade.
    - Check My Oracle Support Document 161818.1: Oracle Database (RDBMS) Releases Support Status Summary for any PSUs that need to be applied. Click on your target release (RELEASE column) in the colored table - click on "Availability and Known issues" or "Known issues and alerts" for your desired target patchset - check the Current Recommended Patches section for the list of PSUs.
    - Check My Oracle Support Document 161818.1: Oracle Database (RDBMS) Releases Support Status Summary for any CPUs that need to be applied. Click on your target release (RELEASE column) in the colored table - click on "Availability and Known issues" or "Known issues and alerts" for your desired target patchset - check the Latest Critical Patch Update section for the list of CPUs. (Only use CPUs if there is a business requirement which prevents the use of PSUs.)
    - For RAC, review My Oracle Support Document 1363369.1 Things to Consider Before Upgrading to 11.2.0.3 Grid Infrastructure/ASM and My Oracle Support Document 1212703.1: Oracle Grid Infrastructure 11.2.0.2 Installation or Upgrade may fail due to Multicasting Requirement

NOTE: PSU application is preferred over CPU application whenever both are available. See My Oracle Support Document 854428.1: Patch Set Updates for Oracle Products for more details on PSUs.

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.

How-To

- Document 228257.1: RMAN 'Duplicate Database' Feature in Oracle9i/Oracle Database 10g
- Document 388431.1: Creating a Duplicate Database on a New Host
- Document 388424.1: How To Create A Production (Full or Partial) Duplicate On The Same Host

- Preserve configuration information, object / system statistics, and performance baselines (see *Prepare and Preserve step, Preserving Configurations and Statistics for the Source Database*)
- Create a test database for the test upgrade. 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:
  - Use Oracle Recovery Manager (RMAN) to duplicate the database. For details see Oracle Database Backup and Recovery Advanced User's Guide 10g Release 2 > Creating and Updating Duplicate Databases with RMAN.
  - 3rd party and home-grown backup and recovery solutions may be used as alternatives to RMAN.
  - Re-create the database
  - Export/Import (if upgrading from Oracle Database 9.2.0.8)
  - Data Pump Export/Import (if upgrading from Oracle Database 10g)

  NOTE: Oracle Data Masking can help you comply with data privacy and protection mandates. With Oracle Data Masking, sensitive information such as credit card or social security numbers can be replaced with realistic values, allowing production data to be safely used for testing.

- Perform a test upgrade using the test database you just created. 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 in case you must start over. 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 step also ensures that your test upgrade can be repeated if necesssary. 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 My Oracle Support Document 466452.1: Best Practices for Load Testing.
- 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.
- Execute the testing process 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 you are ready to perform another upgrade.

Known Issues

This section includes known issues which could be encountered by a wide range of customers during the upgrade process. These issues may prevent a successful upgrade. This is not a complete list of known issues. For additional known issues, please search My Oracle Support. For behavior changes and recommended patches please click on the *Behavior Changes* or *Patches Recommended* tabs.

Notes
• Document 880782.1: ALERT: Oracle 11g Release 2 (11.2) Support Status and Alerts

- Database Issues
  - If moving from Standard Edition to Enterprise Edition see the warning under Oracle Database Upgrade Guide 11g Release 2> Chapter 1 > Moving From Standard Edition to Enterprise Edition of Oracle Database
  - CURSOR_SHARING = SIMILAR no longer limits the number of child cursors that can be created. This could caused performance problems over time which may not appear during testing if the testing is not run long enough to create a huge number of child cursors. See My Oracle Support Document 1169017.1: ANNOUNCEMENT: Deprecating the cursor_sharing = 'SIMILAR' setting for details.
  - Beginning in 10.2.0.2, mutexes to manage the SGA library cache (vs using latches and pins, etc) was introduced. This can lead to Oracle Database 11g Release 2 post-upgrade performance issues. See My Oracle Support Document 727400.1: WAITEVENT: "library cache: mutex X" for details on the "library cache: mutex" wait event and a list of bugs.
  - A large SYS.AUD$ table could cause the upgrade to appear to hang. See My Oracle Support Document 979942.1: Database upgrade appears to have halted at SYS.AUD$ Table.
- RAC or ASM
  - For a detailed list of known issues in Oracle 11g Release 2 for RAC and Grid Infrastructure see My Oracle Support Document 1363369.1 Things to Consider Before Upgrading to 11.2.0.3 Grid Infrastructure/ASM. Since ASM is part of Grid Infrastructure, some of the issues

referenced in this may also apply.

- Operating System Issues
  - AIX increase in memory utilization: refer to My Oracle Support Document 1246995.1: Memory Footprint For Dedicated Server Processes More Than Doubled After 11g Upgrade On AIX Platform for details.
  - 11g Release 2 requires Oracle Solaris 10 Update 6 or greater: refer to My Oracle Support Document 971464.1: FAQ - 11gR2 requires Solaris 10 update 6 or greater for details
  - Oracle recommends that when upgrading to Oracle Database 11g R2 (11.2), you perform an out-of-place installation. However, if you choose to perform an in-place upgrade, follow the steps in the Upgrade Guide carefully. Additional steps have been provided for performing an in-place upgrade on Windows. See the Known Issue When Starting an In-Place Upgrade section of the Oracle Database Upgrade Guide
  - VMware certification: Refer to My Oracle Support Document:942852.1: Oracle VM and VMWare Certification for Oracle Products and Document:249212.1: Support Position for Oracle Products Running on VMWare Virtualized Environments

Defining, implementing and managing a contingency plan is an 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 11g is successful. Planning and practicing for the unexpected issues helps to ensure the upgrade to the new release of Oracle Database 11g is successful.

Prepare

Preparing the database before the upgrade begins helps 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. Refer to My Oracle Support Document 884522.1: How to Download and Run Oracle's Database Pre-Upgrade Utility for details on downloading and running the Pre-Upgrade Utility. This SQL script checks the following:

> IMPORTANT: If you are upgrading the database manually, it is required that the pre-upgrade scripts be run. Ensure that the pre-upgrade scripts are run in the original oracle home. If this step is skipped, the upgrade will terminate early in the process.

1. Database configuration: Determines if the logfiles and datafiles are sized adequately for the upgrade
2. Initialization parameters: Reports which initialization 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.
6. Timezone file version: Reports which file version is used and when/how to upgrade the timezone version.

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

NOTE: $ORACLE_HOME must be set to the source database directory.

Due to timezone changes in Oracle Database 11g Release 2, running the pre-upgrade tool prior to a manual upgrade is now required, otherwise the upgrade script catupgrd.sql will terminate with errors.

If you get a message similar to "Database contains schemas with objects dependent on network packages", then please consult the Oracle Database Upgrade Guide, chapter 4 for further information on Network ACLs.

Oracle recommends gathering data dictionary statistics prior to upgrading the data dictionary. Depending on the size of the data dictionary, the computations can take several hours. Thus, the best practice is to begin the process early enough (such as the night before) to allow plenty of time to collect the statistics before beginning the actual upgrade. See the Oracle Database Upgrade Guide, Appendix B for a detailed script.

NOTE: If upgrading manually from Oracle9i, the SYSAUX tablespace must be created after the new Oracle Database 11g release is started and BEFORE the upgrade scripts are invoked. See My Oracle Support Document 837570.1: Complete Checklist for Manual Upgrades to 11gR2 for details. The SYSAUX tablespace should not be pre-created in the Oracle9i instance. DBUA will create the SYSAUX tablespace.

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, the COMPATIBLE parameter has been set to 11.2, and the database has been restarted, then 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 11.1.0.0.0 incompatible
with ORACLE version 10.2.0.0.0
ORA-00202: control file: '/u01/oradata/B920/control01.ctl'
```

When this error 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 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 10.1.0 or 10.2.0 depending on the setting used before the upgrade until the newly upgraded database performance and functionality is acceptable. When upgrading from Oracle9i Release 2 directly to Oracle Database 11g the minimum setting for COMPATIBLE is 10.1 - so a downgrade from Oracle Database 11g to Oracle9i Release 2 won't be possible. 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 Oracle Database Upgrade Guide 11g.

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 for parameters which have non-default values on your source 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;
```

Oracle recommends removing all underscore parameters and EVENTS unless they are required by an application. If you are unsure about how to handle specific parameters then contact Oracle Support Services.

For Grid Infrastructure Installation: Review Environment Variables

Unset Oracle environment variables. If you have ORA_CRS_HOME set as an environment variable, then unset it before starting an installation or upgrade. You should never use ORA_CRS_HOME as an environment variable.

If you previously had or currently have an installation on your system and you are using the same user account to install this installation, then unset the following environment variables: ORA_CRS_HOME; ORACLE_HOME; ORA_NLS10; TNS_ADMIN.

Refer to the Oracle Grid Infrastructure Installation Guide specific to your platform for more details.

When to Fallback

Prior to the upgrade, you should have a fallback strategy in case performance and functionality is not acceptable and resolving the issues on the upgraded system cannot be done within some agreed upon time. You should answer the following questions to understand when to consider using fallback.

For the most part, only severe performance regressions that cannot be resolved in a timely fashion 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 Service Level Agreements (SLAs)?

For information on the fallback plan and fallback options, see the *Preserving the Database - Fallback Plan* topic 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 11g Installation Guide.***
2. ***Verify that all OS patches and packages are installed as defined in your operating-system specific Oracle Database 11g 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. See Oracle Database SQL Language Reference for the command to alter the tablespace. If any datafile for a tablespace is nearing the limit of 32GB, add a new datafile to that tablespace.***
4. ***You may require larger shared memory pool sizes in some cases. See Oracle Database Reference for information about shared memory initialization parameters. When upgrading from Oracle9i Release 2, the minimum value required for shared pool is 448 MB for 64-bit systems and 224 MB for 32-bit systems. When upgrading from Oracle Database 10g and above, the minimum value required for shared pool is 590 MB for 64-bit systems and 295 MB for 32-bit systems. Computers with large numbers of CPUs may have higher minimum requirements for SGA memory.***
5. ***If you are using Database Upgrade Assistance (DBUA) as the method to upgrade, make sure that there is sufficient OS temp space.***

6. ***Ensure there are no invalid objects in `SYS` and `SYSTEM` user schema.***

Check for invalid objects by performing the following commands:

```
spool invalid_pre.lst
select substr(owner,1,12) owner,
substr(object_name,1,30) object,
substr(object_type,1,30) type, status from
dba_objects where status <>'VALID';
spool off
```

If there are invalid objects you can recompile them by logging in as a SYSDBA user and running the `utlrp.sql` located in the `$ORACLE_HOME/rdbms/admin` directory. This script will attempt to recompile the invalid objects and any dependencies. This script can be run multiple times until all objects have been compiled.

If invalid objects still exist which do not have a name that begins with x_$, check the owner of objects which can't be compiled successfully. In most cases these objects belong to database options which have been installed in earlier releases but do not exist in the current installation anymore. In this case a support request (SR) with Oracle Support should be opened to ask for the necessary scripts to drop these objects.

If valid or invalid objects exists that have a name that begins with x_$, then refer to My Oracle Support Document 361757.1: Invalid x_$ Objects After Upgrade. These views were created by third party applications and point to non-existent or modified x$ tables. Since these are not Oracle created objects, they should be dropped before the upgrade. They cannot be validated or dropped after the upgrade using normal methods.

NOTE: If using DBUA, see Oracle Database Upgrade Guide 11g Release 2 > Chapter 3 > Troubleshooting the Upgrade of Oracle Database > DBUA May Mark Invalid Components with an X Before Entire Upgrade is Done.

NOTE: The Pre-Upgrade Information Tool checks for invalid objects. Refer to My Oracle Support Document 884522.1: How to Download and Run Oracle's Database Pre-Upgrade Utility for details on downloading an running the Pre-Upgrade Utility.

NOTE: To avoid effecting database performance, run the utlrp.sql script during a time when the load on the database is low.

7. ***Verify that all dba_registry components are valid.***

If invalid components exist in dba_registry try the the steps below.

   a. ***Recompile all of the invalid objects with utlrp.sql.***
   b. ***If there are still invalid components after running utlrp.sql, then consult My Oracle Support for other solutions such as Document 472937.1: Information On Installed Database Components and Schemas and Document 753041.1: How to diagnose Components with NON VALID status.***

      Use the following query to identify invalid compontents that may still exist:

```
SQL> select substr(comp_id,1,15) comp_id,
substr(comp_name,1,30) comp_name,
substr(version,1,10) version,
status
from dba_registry
order by modified;
```

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

Please note that using STARTUP UPGRADE automatically disables 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. ***Starting with Oracle Database 10g Release 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. Refer to My Oracle Support Document 884522.1: How to Download and Run Oracle's Database Pre-Upgrade Utility for details on downloading an running the Pre-Upgrade Utility.

```
SELECT grantee
FROM dba_role_privs
WHERE granted_role = 'CONNECT' and
      grantee NOT IN ('SYS', 'OUTLN', 'SYSTEM',
           'CTXSYS', 'DBSNMP',
           'LOGSTDBY_ADMINISTRATOR',
           'ORDSYS', 'ORDPLUGINS',
           'OEM_MONITOR', 'WKSYS',
           'WKPROXY', 'WK_TEST',
           'WKUSER', 'MDSYS',
           'LBACSYS', 'DMSYS', 'WMSYS',
           'EXFSYS', 'SYSMAN',
           'MDDATA', 'XDB', 'ODM',
           'SI_INFORMTN_SCHEMA');
```

10. ***It is mandatory to run the pre-upgrade tool prior to the upgrade process otherwise the upgrade will not continue. The pre-upgrade tool validates whether the source database is ready to be upgraded. Any area that does not meet the requirements must be corrected. Refer to My Oracle Support Document 884522.1: How to Download and Run Oracle's Database Pre-Upgrade Utility for details on downloading an running the Pre-Upgrade Utility.***
11. ***Use Oracle Recovery Manager (RMAN) to take a complete online backup of your database as well as make a backup copy of the following files: init.ora or spfile, password file, and all SQL*Net files (sqlnet.ora, listener.ora, tnsnames.ora, and so on...).***
12. ***Consult the "Upgrading Your Applications" chapter of the upgrade guide for specific instructions regarding applications.***
13. ***Back up the oraInventory and ORACLE_HOME directories.***

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 and the remediation of performance issues that arise once the upgrade is completed. Without pre-upgrade data it will be more difficult to determine why performance characteristics have changed.

Documentation

- Backup and Recovery User's Guide
- Oracle Database Real Application Testing User's Guide 11g
- Oracle Database Performance Tuning Guide 11g Release 2 > Chapter 17 > Managing SQL Tuning Sets

Notes

- Document 560977.1: Real Application Testing Now Available for Earlier Releases
- Document 1268920.1: Real Application Testing:Workload Analyzer
- Document 1287620.1: Database Replay Diagnostic information
- Document 562899.1: TESTING SQL PERFORMANCE IMPACT OF AN ORACLE 9i TO ORACLE DATABASE 10g RELEASE 2 UPGRADE WITH SQL PERFORMANCE ANALYZER
- Document 742644.1: SQL PERFORMANCE ANALYZER 10.2.0.x to 10.2.0.y EXAMPLE

Preserving Performance Baselines and Statistics for the Source Database

Proper performance testing is the 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 current Oracle9i/Oracle Database 10g PRODUCTION, TEST and upgraded PRODUCTION). These baselines help 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 workload performance data captured prior to the upgrade from Oracle9i Release 2/ Oracle Database 10g Release 2 and after completing the upgrade***

to Oracle Database 11g

NOTE: ***SQL Plan Management (SPM) is a freely available new feature in Oracle Database 11g that ensures plan stability and the same plan as in the original (pre-upgrade) database release. With Oracle Database 11g and SPM, the optimizer automatically manages plans and ensures that only verified or known plans are used. SPM allows controlled plan evolution by only using a new plan after it has been verified to perform better than the current plan. Some small amount of preparation is needed to ensure that execution plans are preserved for use after the upgrade, but for best results, use SQL Plan Management as part of your upgrade strategy.***

For more details on SQL Plan Management:

- ***Oracle Database Performance Tuning Guide - Using SQL Plan Management***
- ***SQL Plan Management in Oracle Database 11g***
- ***Document 456518.1: SQL PLAN MANAGEMENT***
- ***Inside the Oracle Optimizer***

In addition to using SPM, Real Application Testing is a paid option that includes Database Replay and SQL Performance Analyzer (SPA). Use of this option is highly recommended for mission critical databases, and for instances where it is important to verify application functionality and performance.

For critical databases and best results, use Database Replay to capture actual production workload in Oracle9i/Oracle Database 10g for replay in an Oracle Database 11g environment. Additionally, for testing SQL execution plans, use SQL Performance Analyzer (SPA) on those same databases. Starting with Oracle9i, an extended SQL trace can be generated to capture SQL execution performance for key workloads for input to SPA. For Oracle Database 10g Release 2, the more efficient incremental cursor cache capture should be used to capture SQL into a SQL Tuning Set for input to SPA. This may require licensing the Real Application Testing option and Oracle Database Diagnostic and Tuning Management Packs, but makes the task of testing upgrades significantly easier and hence is the recommended best practice.

Baselines, Step-by-step

1. ***Choose 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 by either:***
    - ***Using Real Application Testing functionality (if licensed) to capture actual production workloads directly. This will provide the best results for upgrades***
    - ***Scripting the unit tests***

SCRIPTS

- Document 742645.1: Database Replay Command Line Interface (CLI) usage examples/scripts
- Document 787658.1: SCALE_UP_MULTIPLIER: DATABASE CAPTURE AND REPLAY

How-To

- Document 465787.1: Managing CBO Stats during an upgrade to 10g or 11g
- Document 466350.1: Recording Explain Plans before an upgrade to 10g or 11g
- Document 376442.1: Recommended Method for Obtaining 10046 trace for Tuning

Scripts and Tools

- Document 301137.1: OS Watcher
- Document 461053.1: OSWg
- Document 352363.1: LTOM

White Papers

- Maximum Availability Architecture (MAA)
- Document 466996.1: Determining CPU Resource Usage for Linux and Unix
- Document 467018.1: Measuring Memory Resource Usage for Linux and Unix
- SQL Performance Analyzer
- Database Replay
- Siebel on Exadata
- Upgrading from Oracle Database 10g to 11g: What to expect from the Optimizer

4. ***Construct accurate load tests by choosing the mix of activities, number of users, and activity rate that simulates production workloads. Do this using one of the following:***
    - ***Use the Real Application Testing option to capture actual production workloads directly for best results. You will be able to replay these workloads on the TEST system to simulate the production load before and after upgrading.***
    - ***Use a 3rd party load testing tool or scripting to simulate the production workload on the TEST system.***

On the TEST system, perform the following BEFORE upgrading:

1. ***Capture performance baselines for the unit tests. This includes:***
    - ***Execution timing for each activity. We will use this later to see if a job is taking longer after the upgrade.***
    - ***Statspack snapshots: My Oracle Support Document 394937.1 taken 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.***

        NOTE: If you are licensed for the Diagnostic Management Pack, you can capture AWR snapshots before and after the upgrade and then compare them using the AWR Comparison Report ($ORACLE_HOME/rdbms/admin/awrddrpt.sql or for Oracle Database 11g Release 2 with RAC use awrgdrpt.sql). This is an easier way to compare performance metrics.

        CAUTION: Confirm your AWR (or Statspack) retention interval is long enough to ensure your test data is available well after the upgrade.

2. ***Extended SQL tracing: My Oracle Support Document 376442.1 (set event 10046 level 12; if using SQL Performance Analyzer (SPA), refer to My Oracle Support Document 562899.1: TESTING SQL PERFORMANCE IMPACT OF AN ORACLE 9i TO ORACLE DATABASE 10g RELEASE 2 UPGRADE WITH SQL PERFORMANCE ANALYZER for details on enabling SQL trace) Start the trace at the session level just before the activity and end the trace just after the activity. Please note that SQL tracing is expensive and affects the performance of the session being traced.***
    - ***Starting with Oracle9i, this trace may be used in SPA for comparison after the upgrade.***
    - ***If upgrading from Oracle Database 10g and if you are licensed for the Diagnostic and Tuning Management Packs, you can capture the SQL from the cursor cache***

directly and avoid SQL tracing.

3. *Use the Tuning Pack option to capture the SQL Tuning Set (Available starting with Oracle Database 10g Release 2) which can be used by SPA: My Oracle Support Document 562899.1 to compare execution plans and performance of the SQLs after the upgrade.*
4. *Run the load tests and capture the following data:*
    - *If using Real Application Testing, use DB Replay to play back the production load on the TEST system*
    - *Operating system metrics (OS Watcher: My Oracle Support Document 301137.1 is recommended)*
    - *Statspack snapshots: My Oracle Support Document 394937.1 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.*

      ***NOTE:** If you are licensed for the Diagnostic Management Pack, you can capture AWR snapshots before and after the upgrade and then compare them using the AWR Comparison Report. This is an easier way to compare performance metrics.*

      ***CAUTION:** Confirm your AWR (or Statspack) retention interval is long enough to ensure your test data is available well after the upgrade.*

5. *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:

1. *Capture performance baselines for each critical activity separately. This includes:*
    - *For best results use Database Replay to capture peak/ interesting workload periods.*
    - *Gathering execution timing and activity application metrics (e.g, orders shipped per minute) for each activity. You can use this later to see if a job is taking longer after the upgrade.*
    - *Extended SQL tracing: My Oracle Support Document 376442.1 (set event 10046 level 12; if using SQL Performance Analyzer (SPA), refer to My Oracle Support Document 562899.1: TESTING SQL PERFORMANCE IMPACT OF AN ORACLE 9i TO ORACLE DATABASE 10g RELEASE 2 UPGRADE WITH SQL PERFORMANCE ANALYZER for details on enabling SQL trace). Start the trace at the session level just before the activity and end the trace just after the activity.*
        - *Starting with Oracle9i, this trace may be used in SPA for comparison after the upgrade.*
        - *If upgrading from Oracle Database 10g and if you*

are licensed for the Diagnostic and Tuning Management Packs, you can capture the SQL from the cursor cache directly and avoid SQL tracing.

- ***With Real Application Testing (RAT) it is possible to use SPA for further performance analysis on each critical activity. SPA has been enhanced to support Oracle9i/ Oracle Database 10g to Oracle Database 11g upgrades. View My Oracle Support Document 562899.1: TESTING SQL PERFORMANCE IMPACT OF AN ORACLE 9i TO ORACLE DATABASE 10g RELEASE 2 UPGRADE WITH SQL PERFORMANCE ANALYZER for detailed steps on using SPA. View My Oracle Support Document 560977.1: Real Application Testing Now Available for Earlier Releases for backport details.***
- ***A license is needed for Real Application Testing (RAT) in order to use SPA.***

2. ***Use the Tuning Pack option to capture the SQL Tuning Set (Available starting with Oracle Database 10g Release 2) which can be used by SPA to compare execution plans and performance of the SQLs after the upgrade.***
3. ***Capture the following metrics when critical jobs or peak loads are running:***
   - ***Operating system metrics (OS Watcher: My Oracle Support Document 301137.1 is recommended)***
   - ***Use Oracle Database Diagnostic/Tuning Management Packs to capture performance data (recommended)***
     - ***You can manually trigger AWR snapshots using the DBMS_WORKLOAD_REPOSITORY. CREATE_SNAPSHOT procedure. Baselines can also be recorded using the DBMS_WORKLOAD_REPOSITORY. CREATE_BASELINE procedure.***
   - ***If not using Management Packs then use Statspack snapshots: My Oracle Support Document 394937.1 taken just before and just after the workload (at level 7, with low SQL capture thresholds - see My Oracle Support Document 466350.1: Recording Access Path Information Prior to an upgrade to 10g or 11g for details). Additional snapshots may be taken at 1/2 hour intervals.***

     NOTE: If the Oracle9i Release 2/Oracle Database 10g production system is already resource-constrained (CPU, memory, and/or I/O) as seen in the OS Watcher (OSW) or OS Watcher Graph (OSWg) output, then the Oracle Database 11g system will also be resource-constrained, then consider a capacity increase before upgrading. For guidance on interpreting CPU and memory performance diagnostics, please see the COE white papers: Document 466996.1: Determining CPU Resource Usage for Linux and Unix and Document 467018.1: 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)(See My Oracle Support Document:314422.1: Remote Diagnostic Agent (RDA) 4 - Getting Started) on your production database and save the report for future reference. See My Oracle Support Document 465787.1: Managing CBO Stats during an upgrade to 10g or 11g before upgrading from Oracle9i/Oracle Database 10g to Oracle Database 11g to see what should be done with object statistics.***
2. ***Backup Performance-related data***
    - ***Export the PERFSTAT user to preserve statspack data.***
    - ***Export the OUTLN user***
    - ***Export the statspack user***
    - ***Backup collected performance data***
    - ***Export AWR data if licensed to use AWR (Starting in Oracle Database 10g Release 2)***

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

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 initialization parameter file. 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 does not 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 proven, well tested and confirmed to recover the database in case of failure. Make sure that your backup tapes or virtual tape drives are accessible in case they are needed.

Further information on taking full database backups can be found in the Oracle Database Backup and Recovery User's Guide.

As well as taking a backup of the database, you should backup the oraInventory and the ORACLE_HOME so that they can be restored if needed. To perform a backup, copy the oraInventory directory and the ORACLE_HOME (with or without compression). Taking a backup of the current ORACLE_HOME directory also provides a stable installation that can be reinstated should you need to fallback to the pre-upgraded version.

1. ***Fallback Options***

Prior to changing database compatibility

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

OPTIONS	*STEPS*	*CONSIDERATIONS*
Downgrade	1. ***Shutdown database*** 2. ***Downgrade*** 3. ***Restart***	• ***Zero Data Loss*** • ***Database integrity must be in place*** • ***Downgrade process does not restore the database dictionary to the pre-upgrade state.*** • ***Only allows the database to be accessed by the previous version of the software*** • ***Refer to Oracle Database Upgrade Guide***
Streams	1. ***Switch back to your replica***	• ***Zero Data Loss*** • ***Streams requirement*** • ***Refer to Oracle Streams Administrator's Guide*** • ***< 1 minute RTO***
GoldenGate	1. ***Switch back to your replica***	• ***Zero Data Loss*** • ***Streams requirement*** • ***Refer to Oracle GoldenGate Windows and UNIX Administrator's Guide 11g Release 1 Patch Set 1 (11.1.1.1)*** • ***< 1 minute RTO***

Export/ Import (When falling back to Oracle Database 9.2.0.8)	1. *Export or Unload all changes* 2. *Import changes*	• *Zero Data Loss* • *Very time consuming* • *Existence of Read Only tablespaces can reduce work data by skipping those tables* • *Refer to Oracle Database Utilities*
Export/Import Data Pump (when falling back to Oracle Database 10g)	1. *Export or Unload all changes* 2. *Import changes*	• *Zero Data Loss* • *Very time consuming* • *Existence of Read Only tablespaces can reduce work data by skipping those tables* • *Refer to Oracle Database Utilities*
Restore to backup	*Restore, Recover and Activate*	• *Data Loss* • *Refer to Oracle Database Backup and Recovery User's Guide*

After changing database compatibility

After changing database compatibility, you have fewer options.

OPTIONS	STEPS	CONSIDERATIONS
Streams	1. *Switch back to your replica*	• *Zero Data Loss* • *Streams requirement* • *Refer to Oracle Streams Administrator's Guide* • *< 1 minute RTO*
GoldenGate	1. *Switch back to your replica*	• *Zero Data Loss* • *Streams requirement* • *Refer to Oracle GoldenGate Windows and UNIX Administrator's Guide 11g Release 1 Patch Set 1 (11.1.1.1)* • *< 1 minute RTO*

Export/Import Data Pump	1. ***Export or Unload all changes*** 2. ***Import changes***	• ***Zero Data Loss*** • ***Very time consuming*** • ***Existence of Read Only tablespaces can reduce work data by skipping those tables*** • ***Refer to Oracle Database Utilities***
Restore to backup	***Restore, Recover and Activate***	• ***Data Loss*** • ***Refer to Oracle Database Backup and Recovery User's Guide***

You are now ready to upgrade your environment to Oracle Database 11g. 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 Checklist

At this point you should have:

- Reviewed the Oracle Database 11g Upgrade Guide?
- Read the Oracle Database Readme for Oracle Database 11g?
- Read the Oracle Database Installation Guide for your specific Operating System?
- Read the Oracle Administrator Reference 11g for UNIX Based Operating Systems?
- 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 (My Oracle Support Document 884522.1) and resolved any reported errors.

  IMPORTANT : If you are upgrading the database manually, it is required that the pre-upgrade scripts be run. Ensure that the pre-upgrade scripts are run in the original oracle home. If this step is skipped, the upgrade will terminate early in the process.

  - When this script runs, it performs the following tasks:
    - Displays the global database info and checks the compatibility initialization parameter
    - Checks for redo logfile sizes less than 4MB
    - Checks that all tablespaces are checked for adequate space prior to upgrade
    - Checks for objects in the recycle bin and outputs a message requiring it to be purged
    - Displays a list of parameters which need to be corrected prior to upgrade
    - Displays a list of deprecated and obsolete parameters in Oracle Database 11g
    - Displays a list of hidden parameters which should be removed in Oracle Database 11g
    - Displays a list of events defined which should be reviewed
    - Provides a list of component features that will be upgraded
    - Displays a list of all invalid objects and other miscellaneous warnings
    - Displays a warning on what to do if the timezone file version in the db is older or newer than the file shipped with 11gR2

release.

- Verified that the desired OS-Oracle combination is certified by going to My Oracle Support, clicking on the "Certifications" tab, and following the steps to verify your OS and Oracle combination.
- Verified the kernel parameters are set according to the Oracle Database 11g Installation Guide?
- Reviewed and installed recommended operating system patches found in My Oracle Support Document:1265700.1: Oracle Patch Assurance - Data Guard Standby-First Patch Apply?
- Collected object and system (if applicable) statistics and performance baselines from the Oracle9i/Oracle Database 10g production system as described in the *Prepare and Preserve step, Preserving Configurations and Statistics for the Source Database* sections.
- Created a duplicate production environment for Testing Purposes?
- Installed newest patch set(full patch set starting with 11.2.0.2)?
- Applied newest available patch set update(PSU)?
- Applied Critical Patches Updates (CPUs)? (Only use CPUs if there is a business requirement which prevents the use of PSUs)
- Applied recommended (bundled) patches(BP)?
- Applied one-off patches for known issues?
- 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 11g 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 11g

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 occurs in two stages:

1. Perform the following tasks 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, if the upgrade on TEST was successful, perform the following tasks on PRODUCTION:
    - 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

NOTE: Oracle Database 10g and 11g provide enhanced performance diagnostic features that can be used if you are licensed for them. These features include Automatic Workload Repository (AWR), Active Session History (ASH), Automatic Database Diagnostic Monitor (ADDM), SQL Performance Analyzer (SPA), and DB Replay.

Post Upgrade Tasks

After the database has been upgraded, a few additional tasks should be performed before performance is evaluated. These tasks include:

1. Follow procedures in Oracle Database Upgrade Guide 11g - Chapter 4: After Upgrading a Database
2. Review any readme's associated with applied recommended patches and follow instructions provided.
3. Follow directions in My Oracle Support Document 461082.1: Do I Need To Run catcpu.sql After Upgrading A Database?
4. Adjust time zone data in the database to DST V14 or higher. For more information see the Oracle Database Globalization Support Guide.
5. Gather system statistics during a regular workload period.

   ```
   SQL>execute dbms_stats.gather_system_stats('start');
   <<Run for several hours during a regular workload period.>>
   SQL>execute dbms_stats.gather_system_stats('stop');
   ```

   Refer to the Performance Tuning Guide for a listing of all stats.

6. Create fixed table statistics immediately after catupgrd.sql has completed:

   ```
   SQL> execute dbms_stats.gather_fixed_objects_stats;
   ```
   NOTE: This displays a recommendation to remove all hidden or underscore parameters and events from init.ora/spfile.

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.
3. Verify the database component versions and that the status of each component is VALID.

```
SQL> select substr(comp_id,1,15) comp_id,
substr(comp_name,1,30) comp_name,
substr(version,1,10) version,
status
from dba_registry
order by modified;
```

Oracle RAC Only

Notes

- Document 331168.1: Oracle Clusterware consolidated logging in 10gR2/11

Scripts and Tools

- Document 330358.1: CRS 10gR2/11g Diagnostic Collection Guide
- Document 265769.1: Troubleshooting CRS Reboots

HA Documentation

- Oracle Database High Availability Overview

1. Ensure nodes are stable and no evictions occur.
2. Review the Oracle Clusterware, 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 My Oracle Support Document 330358.1: CRS 10gR2/11g Diagnostic Collection Guide

Instability issues are usually caused by bugs or an incorrect configuration (using uncertified components). Ensure you have installed version 11.2.0.X and applied the recommended patches (see the Recommended Patches tab) properly.

Double-check My Oracle Support's Certifications page to be sure you have installed certified 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 11g. There are changes in Oracle Database 11g with regard to the CBO and having proper statistics gathered is essential to getting good performance.

See My Oracle Support Document 465787.1: Managing CBO Stats during an upgrade 10g or 11g after upgrading from Oracle9i or Oracle Database 10g to Oracle Database 11g 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

- Document 369412.1: Resolving High CPU Usage on Oracle Servers

Documentation

- Oracle Database 11g Upgrade Guide > Ch. 4: After Upgrading
- Oracle Performance Tuning Guide, SQL Tuning Overview
- SQL Tuning Advisor
- Oracle Database Real Application Testing User's Guide 11g
- Oracle Database Performance Tuning Guide 11g Release 2 > Chapter 10 > Real-Time SQL Monitoring

How-To

- Document 376442.1: Recommended Method for Obtaining 10046 trace for Tuning
- Document 232443.1: How to Identify Resource Intensive SQL for Tuning
- Document 465787.1: Managing CBO Stats during an upgrade to Oracle Database 10g or 11g
- Document 466350.1: Recording Explain Plans on

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: My Oracle Support Document 394937.1 (or AWR snapshots, included in the Diagnostic Pack) just before and just after the critical job or activity.***
    - ***If using Statspack, 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 a Statspack report (or an AWR report if you are licensed to use it) 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 the new Oracle Database version 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.***
    - ***If upgrading from Oracle Database 10g to 11g, compare AWRs using the AWR Comparison Report (be sure to preserve the Oracle Database 10g baselines before upgrading). The AWR Comparison Report allows thorough comparisons of database performance metrics.***
    - ***If upgrading your RAC system to Oracle Database 11g Release 2, you can use the awrgrpt.sql for a cluster-wide report and awrgdrpt.sql for a cluster-wide comparison report.***

3. ***Extended SQL tracing: My Oracle Support Document 376442.1 (SPA method)***
    - ***SQL Performance Analyzer (SPA) has been enhanced to***

Oracle9i before an upgrade to Oracle Database 10g or 11g
- Document 390374.1: Oracle Performance Diagnostic Guide, Query Tuning

Notes
- Document 247611.1: Known RMAN Performance Problems
- Document 560977.1: Real Application Testing Now Available for Earlier Releases
- Document 562899.1: TESTING SQL PERFORMANCE IMPACT OF AN ORACLE 9i TO ORACLE DATABASE 10g RELEASE 2 UPGRADE WITH SQL PERFORMANCE ANALYZER
- Document 742644.1: SQL PERFORMANCE ANALYZER 10.2.0.x to 10.2.0.y EXAMPLE SCRIPTS
- Document 742645.1: Database Replay Command Line Interface (CLI) usage examples/scripts
- Document 787658.1: SCALE_UP_MULTIPLIER: DATABASE CAPTURE AND REPLAY

Scripts and Tools
- Document 301137.1: OS Watcher
- Document 461053.1: OSWg
- Document 352363.1: LTOM
- Document 224270.1: Trace Analyzer

White Papers

support Oracle9i/Oracle Database 10g to Oracle Database 11g upgrades. Using SPA, now you can load Oracle9i production SQL trace files into an Oracle Database 11g test system, execute the SQLs on an Oracle Database 10g Release 2 test database to capture the post-upgrade performance, and then compare the performance data on an Oracle Database 11g SPA system.

- ***Set event 10046. Refer to My Oracle Support Document 562899.1: TESTING SQL PERFORMANCE IMPACT OF AN ORACLE 9i TO ORACLE DATABASE 10g RELEASE 2 UPGRADE WITH SQL PERFORMANCE ANALYZER for details on enabling SQL trace on the test system. Start the trace at the session level just before the activity and end the trace just after the activity.***
- ***For best results use SPA for performance analysis on each critical activity. View My Oracle Support Document 562899.1: TESTING SQL PERFORMANCE IMPACT OF AN ORACLE 9i TO ORACLE DATABASE 10g RELEASE 2 UPGRADE WITH SQL PERFORMANCE ANALYZER for detailed steps on using SPA and view My Oracle Support Document 560977.1: Real Application Testing Now Available for Earlier Releases for SPA backport details.***

4. ***Extended SQL tracing: My Oracle Support Document 376442.1***
   - ***Set event 10046 at level 12***
   - ***Start the trace at the session level just before the activity and end the trace just after the activity (for details on how to obtain these trace files, see DBMS_MONITOR package).***
   - ***Obtain a TKProf report of the test before and after the upgrade***
   - ***Use Oracle Support's utility, Trace Analyzer: My Oracle Support Document 742645.1 for enhanced reporting of 10046 trace data.***
   - ***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.

- Document 466452.1: Best Practices for Load Testing System Upgrades
- Document 466996.1: Determining CPU Resource Usage for Linux and Unix
- Document 467018.1: Measuring Memory Resource Usage for Linux and Unix
- SQL Performance Analyzer
- Database Replay
- Upgrading from Oracle Database 10g to 11g: What to expect from the Optimizer

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.***
3. ***If you captured the actual production database load using DB Replay, you can replay it on the TEST system after upgrading to Oracle Database 11g. Otherwise, you will need to use a 3rd-party load simulator product or scripts.***
4. ***If you are using the database capture and replay feature in Oracle Database 11g Release 2 you can perform stress testing for query workload using the SCALE_UP_MULTIPLIER parameter. See My Oracle Support Document 787658.1: SCALE_UP_MULTIPLIER: DATABASE CAPTURE AND REPLAY for details.***

Please see My Oracle Support Document 466452.1: 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...):

- ***OS data (sar, vmstat, or OS Watcher: My Oracle Support Document 301137.1)***
- ***Database data (Statspack: My Oracle Support Document 394937.1, AWR) for system-wide evaluation (1/2 hour snapshots, level 7 if statspack), and extended SQL trace: My Oracle Support Document 376442.1 (10046) / TKprof for individual batch jobs or key sessions/activities.***
- ***As mentioned above, AWR comparisons may be performed using awrddrpt.sql or by using awrgdrpt.sql for Oracle Database 11g Release 2 RAC.***
- ***If you are using DB Replay, you only have performance data from pre-upgrade PRODUCTION (obtained during actual use) and post-upgrade TEST (obtained with DB replay) to compare.***

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***
  - ***If you collected pre- and post-upgrade OS data using OS Watcher: My Oracle Support Document 301137.1, you can now graph the data using OSWg: My Oracle Support Document 461053.1 and compare the differences.***
  - ***For guidance on interpreting CPU and memory performance diagnostics, please see the COE white papers: Document 466996.1: Determining CPU Resource Usage for Linux and Unix and Document 467018.1: 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***
  - ***Use the Oracle9i Release 2 statspack data reports during critical activity times and estimate DB time by adding the top 5 or 10 wait times plus the CPU time. On Oracle Database 10g Release 2, you can see the value of DB Time in the Time Model section of the report and can compare Oracle Database 10g to 11g using the AWR Comparison Report.***
  - ***In Oracle Database 11g, collect an AWR report of similar time frame to the Oracle9i Release 2/Oracle Database 10g 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 /Oracle Database 10g Release 2 and Oracle Database 11g, then the upgrade has not regressed the application's performance. Otherwise, see the next section below called, "Resolving Performance Regressions to help identify what has regressed and why.***
- ***DB Replay has graphical output of performance that you can compare (production capture compared to replay)***

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:

- ***At a minimum, OS data (sar, vmstat, or OS Watcher: My Oracle Support Document 301137.1) and instance-wide data (Statspack: My Oracle Support Document 394937.1, AWR) for system-wide evaluation.***
- ***Optionally, session tracing (extended SQL trace on a few selected sessions to compare with similar traces from the pre-upgrade production system)***
- ***Optionally, if upgrading to Oracle Database 10g or higher, Lite Onboard Monitor (LTOM): My Oracle Support Document 352363.1 Profiles (captures OS and database activity over time). This data is usually captured when performance seems to have regressed after the upgrade.***

Pre/Post Upgrade Performance Comparison using System-wide Metrics

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

- ***Compare OS data during periods of similar activity and load for pre-upgrade and post-upgrade***
  - ***If upgrading to Oracle Database 10g or higher, 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: My Oracle Support Document 301137.1, you can now graph the data using OSWg: My Oracle Support Document 461053.1 and compare the differences.***
  - ***For guidance on interpreting CPU and memory performance diagnostics, please see the COE white papers: Document 466996.1: Determining CPU Resource Usage for Linux and Unix and Document 467018.1: 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 or Oracle Database 10g Release 2 PRODUCTION and Oracle Database 11g PRODUCTION statspack (or AWR) reports of similar time frame (otherwise you'll need to scale the resulting values),***

examine the value of DB Time in the Time Model section of the report.

- *If the DB Time is similar between Oracle9i or Oracle Database 10g Release 2 and Oracle Database 11g, then the upgrade has not regressed the application's performance. Otherwise, you will need to identify what has regressed and why.*

NOTE: *Be sure you are comparing similar workloads, otherwise the comparison may give false positive or negative results.*

Pre/Post Upgrade Performance Comparison of Critical Activities using Session Tracing

For a thorough performance comparison, the critical activities that were timed and traced using extended SQL tracing should be compared after the upgrade. This step is considered optional because if the instance-wide comparison looks acceptable and users are satisfied with performance after the upgrade, there is usually no need for this kind of detailed comparison. However, to be more confident of the database's performance after the upgrade, you may want to do this comparison before the database is released to the user community to ensure all critical activities are performing well.

Compare the performance of critical activities on the production database before and after the upgrade.

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. *Extended SQL tracing: My Oracle Support Document 376442.1*
   - *Set event 10046 level 8*
   - *Start the trace at the session level just before the activity and end the trace just after the activity (for details on how to obtain these trace files, see My Oracle Support Document 376442.1: Recommended Method for Obtaining 10046 trace for Tuning and for Oracle Database 10g see DBMS_MONITOR package). DO NOT start a trace while an activity is running as this will capture performance data that is misleading.*
   - *Obtain a TKProf report or TRCANLZR: My Oracle Support Document 224270.1 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). Generally, the best metrics to compare are logical*

reads (consistent gets + current gets), and CPU time.
- ***If there is a difference that needs to be investigated, rerun TKProf and sort the report by whichever metric significantly changed.***

NOTE: ***If you used the SQL Performance Analyzer (SPA) to compare performance between Oracle9i or Oracle Database 10g and Oracle Database 11g in the test system, SPA analysis is complete at this point - there are no remaining SPA steps to do against the Oracle Database 11g production database. View My Oracle Support Document 562899.1: TESTING SQL PERFORMANCE IMPACT OF AN ORACLE 9i TO ORACLE DATABASE 10g RELEASE 2 UPGRADE WITH SQL PERFORMANCE ANALYZER for detailed steps on using SPA and view My Oracle Support Document 560977.1: Real Application Testing Now Available for Earlier Releases for SPA backport details.***

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

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 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 My Oracle Support Document 232443.1: 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: file_system to ASM, single instance to Oracle RAC, etc.?***

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

NOTE: ***SQL Plan Management (SPM) can be used to ensure that only verified or known plans are used during SQL execution. If execution plans were preserved from the original database release, then SPM can be used at this point to ensure that the optimizer chooses those known good plans. If execution plans were not preserved earlier in the upgrade process, then SPM can still provide a measure of stability by taking advantage of features such as the OPTIMIZER_FEATURES_ENABLE parameter to build plans that mimic those produced by earlier versions of the database.***

For more details on SQL Plan Management:

- ***Oracle Database Performance Tuning Guide - Using SQL Plan Management***
- ***SQL Plan Management in Oracle Database 11g***
- ***Document 456518.1: SQL PLAN MANAGEMENT***
- ***Inside the Oracle Optimizer***

In addition to preserving a base level of performance by ensuring plan stability after the upgrade, you can improve performance by tuning the queries or execution blocks present in your application. Use one of the following methods to tune the SQL:

1. ***If you are licensed for the Tuning Pack, use the SQL Tuning Advisor to tune the SQL.***
2. ***Real-time SQL monitoring is available in 11g for long-running statements (more than 5 seconds of CPU or I/O in a single execution). This capability allows you to see which steps of the execution plan are taking the most time, as well as other metrics such as number of rows processed and amount of logical reads performed. SQL monitoring will also allow you to investigate parallel execution queries.***
3. ***You might also find it helpful to use Oracle Support's SQLT script (See My Oracle Support Document 215187.1: SQLT (SQLTXPLAIN) - Tool that helps to diagnose SQL statements performing poorly) to gather comprehensive information about your query. SQLT output will be requested by Oracle Support engineers if you require assistance.***

4. ***Review possible causes of plan change***
    - ***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 My Oracle Support Document 466350.1: Recording Access Path Information Prior to an upgrade to 10g or 11g for details on how to do this.***
    - ***Investigate what changed between plans (and review the Behavior Changes tab); See the table below for typical causes and solutions.***
    - ***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 staitistics collection method consider using AUTO_SAMPLE_SIZE to take advatange of the new hash-based statistics gathering algorithm in Oracle Database 11g
Rule-based Optimizer to Cost-based Optimizer Migration	See My Oracle Support Document 222627.1: Migrating to the Cost-Based Optimizer and Upgrading from Oracle Database 9i to 10g: What to expect from the 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***
- ***Document 390374.1: Oracle Performance Diagnostic Guide, Query Tuning***

3. ***If Regression is Due to Something Else***

1. ***Consult the Oracle Performance Tuning Guide and My Oracle Support Document 390374.1: 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 11.2.0.x from 10.2.0.x or 9.2.0.x. A discussion on downgrading is found in the *Prepare and Preserve* step and in the Oracle Database Upgrade Guide - Downgrading a Database.

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

Logging an SR After an Upgrade

Oracle Support will generally request OS data (OSWatcher), AWR or Statspack data, and 10046/TKProf/TRCANLZR. Oracle may also request SQLT data if you've isolated the issue to a single query. You will help expedite the SR if you upload this data at the time you log the SR.

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

Document 210014.1 How to Log a Good Performance Service Request

Document 339834.1 Gathering Information for RDBMS Performance Tuning issue

Document 68735.1 Diagnostics for Query Tuning Problems

Document 166650.1 Working Effectively With Global Customer Support

Behavior Changes

This section documents important changes in behavior between Oracle9i Release 2 (9.2)/Oracle Database 10g and Oracle Database 11g. 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 11g. For a complete list of all new features introduced in Oracle Database 11g, see the Oracle Database New Features Guide 11g

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 Oracle9i Release 2/Oracle Database 10g to Oracle Database 11g.

Architecture

Optimal Flexible Architecture (Oracle9i to Oracle Database 10g Change)

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 (Oracle9i to Oracle Database 10g Change)

Parallel Query (PQ) has been enhanced in Oracle Database 10g Release 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 (Oracle9i to Oracle Database 10g Change)

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 (Oracle9i to Oracle Database 10g Change)

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 My Oracle Support Document 345048.1: 'Group By' Does Not Guarantee a Sort Without Order By Clause In 10g and Above.

New Background Processes (Oracle Database 10g to 11g Change)

The following are the new database background process that you could be running depending upon the features being used.

- DBRM (database resource manager) process is responsible for setting resource plans and other resource manager related tasks.
- DIA0 (diagnosability process 0) (only 0 is currently being used) is responsible for hang detection and deadlock resolution.
- EMNC (event monitor coordinator) is the background server process used for database event management and notifications.
- FBDA (flashback data archiver process) archives the historical rows of tracked tables into flashback data archives. Tracked tables are tables which are enabled for flashback archive. When a transaction containing DML on a tracked table commits, this process stores the pre-image of the rows into the flashback archive. It also keeps metadata on the current rows.
- FBDA is also responsible for automatically managing the flashback data archive for space, organization, and retention and keeps track of how far the archiving of tracked transactions has occurred.
- GTX0-j (global transaction) processes provide transparent support for XA global transactions in an Oracle RAC environment. The database autotunes the number of these processes based on the workload of XA global transactions. Global transaction processes are only seen in an Oracle RAC environment
- GEN0 (general task execution process) performs required tasks including SQL and DML
- GMON maintains disk membership in ASM disk groups.
- MARK marks ASM allocation units as stale following a missed write to an offline disk.
- SMCO (space management coordinator) process coordinates the execution of various space management related tasks, such as proactive space allocation and space reclamation. It dynamically spawns slave processes (Wnnn) to implement the task.
- VKTM (virtual keeper of time) is responsible for providing a wall-clock time (updated every second) and reference-time counter (updated every 20 ms and available only when running at elevated priority
- VKRM (virtual scheduler for resource manager process) manages the CPU scheduling for all managed Oracle processes. The process schedules managed processes in accordance with an active resource plan.

Optimizer

The following are some of the important behavior changes in the Optimizer and statistics areas between Oracle9i and Oracle Database 11g. The optimizer development group focused predominately on plan stability and faster statistic gathering techniques in Oracle Database 11g Release 2.

Notes

- Document 295819.1: Upgrading from 9i to 10g - Potential Query Tuning Related Issues
- Upgrading from Oracle Database 9i to 10g: What to expect from the Optimizer

Initialization Parameters (Oracle9i to Oracle Database 10g Change)

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 use to calculate statistics than were used in Oracle9i databases.

3. ***Optimizer_secure_view_merging***

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 prevents 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.

Initialization Parameters (Oracle Database 10g to Oracle Database 11g Change)

1. ***OPTIMIZER_USE_INVISIBLE_INDEXES***

OPTIMIZER_USE_INVISIBLE_INDEXES was introduced in Oracle Database 11g Release 1. This parameter enables or disables the use of invisible indexes. The default value for OPTIMIZER_USE_INVISIBLE_INDEXES is FALSE.

2. ***OPTIMIZER_USE_PENDING_STATISTICS***

OPTIMIZER_USE_PENDING_STATISTICS was introduced in Oracle Database 11g Release 1. This parameter specifies whether or not the optimizer uses pending statistics when compiling SQL statements. The default value for OPTIMIZER_USE_PENDING_STATISTICS is FALSE.

3. ***OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES***

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES was introduced in Oracle Database 11g Release 1. This parameter enables or disables the automatic recognition of repeatable SQL statements, as well as the generation of SQL plan baselines for such statements. The default value for OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES is FALSE.

4. ***OPTIMIZER_USE_SQL_PLAN_BASELINES***

OPTIMIZER_USE_SQL_PLAN_BASELINES was introduced in Oracle Database 11g Release 1. This parameter enables or disables the use of SQL plan baselines stored in SQL Management Base. When enabled, the optimizer looks for a SQL plan baseline for the SQL statement being compiled. If one is found in SQL Management Base, then the optimizer will cost each of the baseline plans and pick one with the lowest cost. The default value for OPTIMIZER_USE_SQL_PLAN_BASELINES is TRUE.

Cost Based Transformations (Oracle9i to Oracle Database 10g Change)

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 (Oracle9i to Oracle Database 10g Change)

The rule-based optimizer (RBO) is not supported in Oracle Database 10g. Oracle Database 10g only supports the cost-based optimizer (CBO). See My Oracle Support Document 189702.1: Rule Based Optimizer is to be Desupported in Oracle10g for more detail.

Automatic Optimizer Statistics Collection (Oracle9i to Oracle Database 11g Change)

Oracle Database 10g Release 2

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;
/
```

Oracle Database 11g

In Oracle Database 11g Automatic optimizer statistics collection runs as part of the automated maintenance tasks infrastructure (AutoTask) and is enabled by default to run in all predefined maintenance windows. The AutoTask statistics collection replaces the Oracle Database 10*g* GATHER_STATS_JOB. Automatic optimizer statistics collection is enabled by default. If for some reason automatic optimizer statistics collection needs to be disabled, you can disable it using the DISABLE procedure in the DBMS_AUTO_TASK_ADMIN package.

```
BEGIN
DBMS_AUTO_TASK_ADMIN.DISABLE(
client_name => 'auto optimizer stats collection',
operation => NULL,
window_name => NULL);
END;
```

To re-enable automatic optimizer statistics collection, you can enable it using the ENABLE procedure in the DBMS_AUTO_TASK_ADMIN package

```
BEGIN
DBMS_AUTO_TASK_ADMIN.ENABLE(
client_name => 'auto optimizer stats collection',
operation => NULL,
window_name => NULL);
END;
```

You can query the dba_autotask_client and dba_autotask_job_history to find out the details of Automatic Optimizer Statistics Collection.

```
select client_name,status,TOTAL_CPU_LAST_7_DAYS
from dba_autotask_client where
client_name like 'auto optimizer stats collection';

select CLIENT_NAME,WINDOW_NAME,WINDOW_START_TIME,JOB_STATUS
FROM dba_autotask_job_history where
client_name like 'auto optimizer stats collection';
```

NOTE: There was no default automatic statistics gathering job in Oracle9i.

Changes in default values for DBMS_STATS parameters (Oracle9i to Oracle Database 11g Change)

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_PRECENT	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.

In 11g the behavior for the default value for ESTIMATE_PERCENT has changed. In Oracle Database 11g the sampling algorithm has been completely rewritten. The new algorithm is hash based and provides deterministic statistics, which have the accuracy of computing the statistics but with the speed of a 10% sample. This new algorithm is used when estimate_percent is AUTO_SAMPLE_SIZE (default) in any of the DBMS_STATS.GATHER_*_STATS procedures.

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 (Oracle9i to Oracle Database 10g Change)

In Oracle9i system statistics were introduce 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 systems 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 over ride these initial values. To gather system statistics you can use DBMS_STATS.GATHER_SYSTEM_STATS.

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 Data Dictionary Tables (Oracle9i to Oracle Database 10g Change)

Because 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 data dictionary tables. During the upgrade process Oracle automatically gathers statistics on the data dictionary tables. These statistics are 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 data 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 (Oracle9i to Oracle Database 10g Change)

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_OBJECTS_STATS.

```
BEGIN
DBMS_STATS.GATHER_FIXED_OBJECTS_STATS;
END;
/
```

Statistics on Indexes (Oracle9i to Oracle Database 10g Change)

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 (Oracle9i to Oracle Database 10g Change)

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.

```
BEGIN
DBMS_STATS.RESTORE_TABLE_STATS('HR','EMPLOYEES');
END;
/
```

DBMS_STATS AND PARALLEL DEGREE (Oracle9i to Oracle Database 10g Change)

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 (Oracle9i to Oracle Database 10g Change)

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 re-gather 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. Use the new dbms_stats.set_table_pref to set a specific value for the method_opt parameter. In

this example, histograms will continue to be automatically created on the SALES table except for the column PROD_ID.

```
BEGIN
DBMS_STATS.SET_TABLE_PREFS(('SH', 'SALES',
METHOD_OPT=>'FOR ALL COLUMNS SIZE AUTO FOR COLUMNS SIZE 1 PROD_ID');
END;
/
```

NOTE: For additional information on how the optimizer handles bind variables, see the behavior change titled "Adaptive Cursor Sharing" under the "Optimizer" section.

Cost Based Transformations (Oracle Database 10g to 11g Change)

In Oracle Database 11g three new enhancements have been made to the cost based transformations. The three enhancements are turned on by default and are controlled by the optimizer_features_enabled initialization parameter.

- Null aware anti join

This Feature optimizes queries which have a where clause predicate involving a NOT IN or ALL operator on a column which is NULLABLE. This feature is to improve the performance of this type of query on large dataset. Take for example the following query involving two tables T1 and T2,

SELECT ....
FROM T1
WHERE T1.x NOT IN (select T2.y where ....);

The T2.y column is a NULLABLE column, so in Oracle Database 10g we cannot unnest the subquery thus the only possible execution plan is a cartesian product. In Oracle Database 11g we can use the new null aware anit-join transformation, thus enabling us to unnest the subquery, which means we can use a Nested Loop join, a Hash Join or a Sort Merge Join.

The usage of the null aware anti join feature can be identified in the explain plan by the follow type of operators.

HASH JOIN RIGHT ANTI NA| <--NA means Null Aware

ANTI SNA | <- SNA means Single Null-Aware Anti-Join

Example

select count(*)
from emp
where mgr not in (select mgr from emp);

11.1.0.7.0 Execution Plan

```
Execution Plan
----------------------------------------------------------
Plan hash value: 54517352


-------------------------------------------------------------------------------
| Id| Operation           | Name| Rows| Bytes| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
| 0 | SELECT STATEMENT    |     |   1 |    8 |     7  (15)| 00:00:01 |
| 1 |  SORT AGGREGATE     |     |   1 |    8 |            |          |
|*2 |   HASH JOIN ANTI NA |     |  12 |   96 |     7  (15)| 00:00:01 |
| 3 |    TABLE ACCESS FULL| EMP |  14 |   56 |     3   (0)| 00:00:01 |
| 4 |    TABLE ACCESS FULL| EMP |  14 |   56 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------
```

10.2.0.4.0 Execution Plan

```
Execution Plan
----------------------------------------------------------
Plan hash value: 492197985


-------------------------------------------------------------------------------
| Id| Operation           | Name| Rows| Bytes| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
| 0 | SELECT STATEMENT    |     |   1 |    4 |    21   (0)| 00:00:01 |
| 1 |  SORT AGGREGATE     |     |   1 |    4 |            |          |
|*2 |   FILTER            |     |     |      |            |          |
| 3 |    TABLE ACCESS FULL| EMP |  14 |   56 |     3   (0)| 00:00:01 |
|*4 |    TABLE ACCESS FULL| EMP |   2 |    8 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------
```

- Join Predicate Pushdown

Join predicate pushdown is now available for queries with group-by, distinct, semi/anti-joined view.

Oracle Database 10g introduced the cost based transformation called predicate push down. Predicate push down enables queries that have a join predicate between a table and a view to use a index based nested loop join by pushing the join predicate inside the view. Imagine you have a query, which contains a view V, and a table T. The query also has a join predicate of T.x = V.y. Prior to Oracle Database 10g we could only use two join methods, a hash join or a sort merge join. However, with the introduction of predicate push down we can push the join predicate into the view so the join becomes T.x = T2.y (where T2 is the table inside view V which has the column y in it) then if there is an index present we can do an nested loops join. In Oracle Database 11g we have extended the predicate push down capabilities to include queries, which contain group by, distinct, anti-join and semi-joins. The usage of the new predicate push down feature can be identified in the explain plan by the following operator

VIEW PUSHED PREDICATE

Example

```
select p.prod_name, p.prod_desc, v.qu
from products p,
(select s.prod_id, sum(quantity_sold) qu
```

```
from sales s
group by prod_id) v
where v.prod_id = p.prod_id
and p.supplier_id = 12;
```

11.1.0.7.0 Execution Plan

```
-----------------------------------------------------------------------------------------------------
| Id| Operation                              | Name        | Rows  | Bytes |Cost (%CPU)| Time    |
-----------------------------------------------------------------------------------------------------
| 0 | SELECT STATEMENT                       |             |     1 |    78 |  420   (0)| 00:00:06
| 1 |  NESTED LOOPS                          |             |     1 |    78 |  420   (0)| 00:00:06
|*2 |   TABLE ACCESS FULL                    | PRODUCTS    |     1 |    65 |    3   (0)| 00:00:01
| 3 |   VIEW PUSHED PREDICATE                |             |     1 |    13 |  417   (0)| 00:00:06
|*4 |    FILTER                              |             |       |       |           |         |
| 5 |     SORT AGGREGATE                     |             |     1 |     7 |           |         |
| 6 |      PARTITION RANGE ALL               |             | 12762 | 89334 |  417   (0)| 00:00:06
| 7 |       TABLE ACCESS BY LOCAL INDEX ROWID| SALES       | 12762 | 89334 |  417   (0)| 00:00:06
| 8 |        BITMAP CONVERSION TO ROWIDS     |             |       |       |           |         |
|*9 |         BITMAP INDEX SINGLE VALUE      | SALES_P_BIX |       |       |           |         |
-----------------------------------------------------------------------------------------------------
```

10.2.0.4 Execution Plan

```
-----------------------------------------------------------------------------------------------
| Id| Operation                              | Name        | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
| 0 | SELECT STATEMENT                       |             |    36 |  4824 |   424   (1)| 00:00:06 |
| 1 |  HASH GROUP BY                         |             |    36 |  4824 |   424   (1)| 00:00:06 |
| 2 |   TABLE ACCESS BY LOCAL INDEX ROWID| SALES       | 12762 | 89334 |   422   (0)| 00:00:06 |
| 3 |    NESTED LOOPS                       |             | 12762 |  1670K|   422   (0)| 00:00:06 |
|*4 |     TABLE ACCESS FULL                  | PRODUCTS    |     1 |   127 |     3   (0)| 00:00:01 |
| 5 |     PARTITION RANGE ALL                |             |       |       |            |          |
| 6 |      BITMAP CONVERSION TO ROWIDS       |             |       |       |            |          |
|*7 |       BITMAP INDEX SINGLE VALUE        | SALES_P_BIX |       |       |            |          |
-----------------------------------------------------------------------------------------------
```

Multi level Push predicate

Join predicate push down has also been enhanced to work on a view inside of the view.

- Group By Placement

This new transformation allows the optimizer to rewrite queries in order to minimize the number of rows necessary for a join. The rewrite works by placing the group by operator earlier in the execution plan so less rows are necessary for a join operation. For example if you had the following query

```
SELECT sum(T1.x), T2.y
FROM T1,T2
WHERE T1.x = T2.z
GROUP BY T2.y;
```

The optimizer transforms the query into the following:

```
SELECT sum(V.sumv), T2.y
FROM T2, (select sum(T1.x) as sumv from T1 group by T2.z)V
WHERE V.z=T2.z;
```

Here the group by happens inside the view thus reducing the number of rows coming out of the view into the join with T2.

For example:

```
select prod_name, prod_desc, sum(quantity_sold)
from products p, sales s
where p.prod_id = s.prod_id
group by prod_name, prod_desc;
```

11.1.0.7.0 Execution Plan

```
-------------------------------------------------------------------------------
| Id| Operation               | Name     | Rows | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
| 0 | SELECT STATEMENT        |          |   72 |  5400 |   557  (14)| 00:00:07 |
| 1 |  HASH GROUP BY          |          |   72 |  5400 |   557  (14)| 00:00:07 |
|*2 |   HASH JOIN             |          |   72 |  5400 |   556  (13)| 00:00:07 |
| 3 |    VIEW                 | VW_GBC_5 |   72 |  1224 |   552  (13)| 00:00:07 |
| 4 |     HASH GROUP BY       |          |   72 |   792 |   552  (13)| 00:00:07 |
| 5 |      PARTITION RANGE ALL|          |  918K|  9870K|   497   (4)| 00:00:06 |
| 6 |       TABLE ACCESS FULL | SALES    |  918K|  9870K|   497   (4)| 00:00:06 |
| 7 |    TABLE ACCESS FULL    | PRODUCTS |   72 |  4176 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------
```

The Oracle Database 11g plan shows a view was created (ID 3) and a new group by was added inside the view (ID4).

10.2.0.4 Execution Plan

```
-------------------------------------------------------------------------------------
| Id| Operation             | Name   | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
| 0 | SELECT STATEMENT      |        |  3565 |   226K|       |  5603   (2)| 00:01:08 |
| 1 |  HASH GROUP BY        |        |  3565 |   226K|    67M|  5603   (2)| 00:01:08 |
|*2 |   HASH JOIN           |        |   918K|    56M|       |   507   (5)| 00:00:07 |
| 3 |    TABLE ACCESS FULL  |PRODUCTS|    72 |  4176 |       |     3   (0)| 00:00:01 |
| 4 |    PARTITION RANGE ALL|        |   918K|  6281K|       |   497   (4)| 00:00:06 |
| 5 |     TABLE ACCESS FULL | SALES  |   918K|  6281K|       |   497   (4)| 00:00:06 |
-------------------------------------------------------------------------------------
```

Pruning using bloom filtering (Oracle Database 10g to 11g Change)

In Oracle Database 11g Partition Pruning now uses bloom filtering instead of subquery pruning. While subquery was activated on cost based decision and consumed internal (recursive) resources, pruning based on bloom filtering is activated all the time without consuming additional resources. Bloom filtering gives better performance with large sets.

The usage of the new feature pruning using bloom filtering can be identified in the explain plan by the following operator.

PART JOIN FILTER CREATE :BF0000

The use :BF0000 indicates pruning using bloom filtering.

Pruning Execution Plan

```
-------------------------------------------------------------------------
| Id  | Operation                          | Name      | Pstart| Pstop |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |           |       |       |
|   1 |  PX COORDINATOR                    |           |       |       |
|   2 |   PX SEND QC (RANDOM)              | :TQ10002  |       |       |
|*  3 |    FILTER                          |           |       |       |
|   4 |     HASH GROUP BY                  |           |       |       |
|   5 |      PX RECEIVE                    |           |       |       |
|   6 |       PX SEND HASH                 | :TQ10001  |       |       |
|   7 |        HASH GROUP BY               |           |       |       |
|*  8 |         HASH JOIN                  |           |       |       |
|   9 |          PART JOIN FILTER CREATE   | :BF0000   |       |       |
|  10 |           PX RECEIVE               |           |       |       |
|  11 |            PX SEND PARTITION (KEY) | :TQ10000  |       |       |
|  12 |             PX BLOCK ITERATOR      |           |       |       |
|  13 |              TABLE ACCESS FULL     | CUSTOMERS |       |       |
|  14 |          PX PARTITION HASH JOIN-FILTER|        |:BF0000|:BF0000|
|* 15 |           TABLE ACCESS FULL        | SALES     |:BF0000|:BF0000|
-------------------------------------------------------------------------
```

Distinct elimination (Oracle Database 10g to 11g Change)

Starting with Oracle Database 11g optimizer will go through the query blocks of a select statement and check if distinct can be removed
from the query block, it will remove the distinct if the query block is guaranteed to return distinct rows without it. This feature is applicable only for SELECT statements. If distinct has been eliminated by optimizer than you will not see steps like HASH UNIQUE / SORT UNIQUE in the explain plan.

Example

select distinct empno from emp;

11.1.0.7.0 Execution Plan

```
-------------------------------------------------------------------------
| Id  | Operation        | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT |        |    14 |    56 |     1   (0)| 00:00:01 |
|   1 |  INDEX FULL SCAN | PK_EMP |    14 |    56 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------
```

10.2.0.4 Execution Plan

```
-------------------------------------------------------------------------------
| Id  | Operation          | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |        |    14 |    56 |     2  (50)| 00:00:01 |
|   1 |  SORT UNIQUE NOSORT|        |    14 |    56 |     2  (50)| 00:00:01 |
|   2 |   INDEX FULL SCAN  | PK_EMP |    14 |    56 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------
```

Extended Statistics (Oracle Database 10g to 11g Change)

In real-world data there is often a correlation between two or more columns in a table. For example, job title and salary are related; the VP of a company is likely to earn a lot more than the janitor does or car make and price; a BMW is likely to be a lot more expensive than a Honda. Up until now the optimizer has had no way of knowing that these relationships exist between the columns in a table. So when a query was executed against the table with multiple, single column predicates, it was impossible for the optimizer to calculate the correct selectivity of these predicates as it had no way of knowing if the columns were related or not and it only had individual column statistics to work with. It has also been extremely hard for the optimizer to calculate the correct selectivity for a column that has a function apply to it. For example UPPER (surname)='SMITH'.

In Oracle Database 11g we now provide a mechanism to collect statistics on a group of columns or expressions. You can either specify these columns if you already know which columns are always used together in a query (or have a special correlation) or you can let the optimizer automatically detect these column groups from usage statistics, the same way Oracle determine which columns need histograms in Oracle Database 10g. By gathering statistics on columns as a group (column group), the optimizer will now have a more accurate cardinality estimates for that group, instead of having to generate the value based on the individual column statistics.

Column group statistics include number of distinct values, number of nulls, frequency histograms and density.

The optimizer can also use expression statistics gathered on an expression.

Adaptive Cursor Sharing (Oracle Database 10g to 11g Change)

Prior to Oracle Database 11g when a SQL statement with bind variables was issued for the first time it would be hard parsed and a cost-based plan would be generated. In order to generate the most accurate plan possible the optimizer would 'peek' at the actual literal value of the bind variable and generate a plan based on this value. All subsequent executions of this statement would use this execution plan. This approach worked fine for situations where regardless of the bind value the same plan would be generated. However, it didn't work when there was a data skew in the column which the bind variable referred to. In that case depending on the value passed at the initial hard-parse you would get a plan that would suit most values that would be past but not all or if you were very unlucky you would get a plan that would suit only a small number of values and not suit the majority. One execution plan was not always enough to guarantee good performance.

With Oracle Database 11g the optimizer has the ability to peek at the literal value each time the statement is executed and generate multiple plans for a single statement subsequent bind values require different plans. This way we can use the plan that is best suited to each value.

So how exactly does it work? Just like prior release the optimizer will peek at the first literal value for the bind at hard-parse and generate a cost based plan, based on that value. Then the database will observe the cursor for a while, to see what type of values are passed and if a different plan would be better for some of these values. If the plan does need to be recalculated, the cursor is marked as 'Bind-Sensitive' and we specify a selectivity range for our current plan. As the statement continues to be executed we keep checking the selectivity of the new bind values with that of the existing range. If the new values selectivity falls within the existing range it will use the original plan. But if it's selectivity is greatly different from the original range, a new plan will be generated and the cursor will be marked 'Bind-Aware'. 'Bind-Aware' means there are multiple plans for this statement depending on the selectivity value of the bind variable.

The benefit of using multiple execution plans outweighs the parse time and memory usage overhead.

A new view V$SQL_CS_STATISTICS summarizes the information that the optimizer collects to make this decision. For a sample of executions, it keeps track of rows processed, buffer gets, and CPU time. The use of this feature may lead to more than 1 child cursor for query with bind variables, but there is a limit to the number of child cursor optimizer generates.

Native implementation for full outer joins (Oracle Database 10g to 11g Change)

Before Oracle Database 11g, ANSI full outer joins were converted into a UNION ALL query with two branches, one branch consist of LEFT OUTER JOIN AND other branch consist of NOT EXIST subquery. Oracle Database 11g introduced a native support for hash full outer joins. This improves the performance of a full outer join.

Example

```
select a.*, b.* from emp a
full outer join DEPT b
on a.DEPTNO = b.DEPTNO;
```

11.1.0.7.0 Execution Plan

```
-------------------------------------------------------------------------------
| Id  | Operation             | Name     | Rows  | Bytes | Cost (%CPU)| Time
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |          |    15 |  1755 |     7  (15)| 00:00:01
|   1 |  VIEW                 | VW_FOJ_0 |    15 |  1755 |     7  (15)| 00:00:01
|*  2 |   HASH JOIN FULL OUTER|          |    15 |   855 |     7  (15)| 00:00:01
|   3 |    TABLE ACCESS FULL  | DEPT     |     4 |    80 |     3   (0)| 00:00:01
|   4 |    TABLE ACCESS FULL  | EMP      |    14 |   518 |     3   (0)| 00:00:01
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
2 - access("A"."DEPTNO"="B"."DEPTNO")
```

10.2.0.4.0 Execution Plan

```
-------------------------------------------------------------------------------
| Id  | Operation            | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |      |    15 |  1755 |    21   (5)| 00:00:01 |
|   1 |  VIEW                |      |    15 |  1755 |    21   (5)| 00:00:01 |
|   2 |   UNION-ALL          |      |       |       |            |          |
|*  3 |    HASH JOIN OUTER   |      |    14 |  1638 |    11  (10)| 00:00:01 |
|   4 |     TABLE ACCESS FULL| EMP  |    14 |  1218 |     5   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL| DEPT |     4 |   120 |     5   (0)| 00:00:01 |
|*  6 |    HASH JOIN ANTI    |      |     1 |    43 |    11  (10)| 00:00:01 |
|   7 |     TABLE ACCESS FULL| DEPT |     4 |   120 |     5   (0)| 00:00:01 |
|   8 |     TABLE ACCESS FULL| EMP  |    14 |   182 |     5   (0)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
3 - access("A"."DEPTNO"="B"."DEPTNO"(+))
6 - access("A"."DEPTNO"="B"."DEPTNO")
```

You can see that in Oracle Database 11g the query has used "HASH JOIN FULL OUTER" and only scanned the table EMP and DEPT once. In Oracle Database 10g Release 2, UNION-ALL is done and the tables EMP AND DEPT are scanned twice. So the use of Native implementation for full outer joins improves the performance.

DBMS_STATS (Oracle Database 10g to 11g Change)

Starting with Oracle Database 11g DBMS_STATS uses a new hash based algorithm when using AUTO_SAMPLE_SIZE for ESTIMATE_PERCENT. Prior to Oracle Database 11g, dbms_stats uses a row sampling based algorithm. The new approximate number of distinct value method provide an efficient and accurate method for gathering number of distinct values and other statistics for columns using a hash based algorithm. The approximate NDV technique is used when you invoke a procedure from DBMS_STATS with ESTIMATE_PERCENT gathering option set to AUTO_SAMPLE_SIZE, which is the default value. The row sampling based algorithm will be used for collection of number of distinct values if you specify any value other than AUTO_SAMPLE_SIZE. This preserves the old behavior when you specify sampling percentage. The use of AUTO_SAMPLE_SIZE results in more accurate statistics comparable to that of estimate_percent set to 100, but taking less time as compared to used by dbms_stats when estimate_percent set to 100. There are new features in dbms_stats, using those could also change the statistics gathered by dbms_stats and hence the query plans of a sql may change as compared to earlier versions.

These features are:

- Multicolumn Statistics
- Expression Statistics
- Statistic Preferences

New options are also available in Oracle Database 11g for Statistic Preferences in dbms_stats which where not present in earlier versions.

DBMS_STATS.SET_TABLE_PREF
DBMS_STATS.SET_DATABASE_PREFS
DBMS_STATS.SET_GLOBAL_PREFS

PARAMETER DEFAULT VALUE IN 11G
PUBLISH TRUE

STALE_PERCENT 10
INCREMENTAL FALSE

For more details on these features, see the Oracle Database documentation set.

Comparing Statistics

When it comes to deploying a new application or application module it is standard practice to test and tune the application in a test environment before it is moved to production. However, even with testing it's possible that SQL statements in the application will have different execution plans in production then they did on the test system. One of the key reasons an execution plan can differ from one system to another (from test and production) is because the optimizer statistics on each system are different. In Oracle Database 11g, the DIFF_TABLE_STATS_* function can be used to compare statistics for a table from two different sources. The statistics can be from:

- A user statistics table and current statistics in the data dictionary
- A single user statistics table containing two sets of statistics that can be identified using statids
- Two different user statistics tables
- Two points in history

The function also compares the statistics of the dependent objects (indexes, columns, partitions). The function displays statistics for the object(s) from both sources if the difference between the statistics exceeds a certain threshold (%). The threshold can be specified as an argument to the function; the default value is 10%. The statistics corresponding to the first source are used as the basis for computing the differential percentage.

Stored outlines (Oracle Database 10g to 11g Change)

Oracle highly recommends the use of SQL plan baselines instead of the stored outlines. With Oracle Database 11g using the SQL Plan Management (SPM) the optimizer automatically manages plans and ensures that only verified or known plans are used. SQL Plan Management allows controlled plan evolution by only using a new plan after it has been verified to perform better than the current plan. You should also use SQL Plan Management as part of your upgrade strategy. Please see the best practices section of this Upgrade Companion for more information.

For more details on SQL Plan Management review:
Oracle Database Performance Tuning Guide - Using SQL Plan Management
SQL Plan Management in Oracle Database 11g
Document.456518.1: SQL PLAN MANAGEMENT
Inside the Oracle Optimizer

New density calculation (Oracle9i to Oracle Database 10g)

There is a new algorithm to calculate density in Oracle Database 10g Release 2 when histograms are present on the columns. This shows up in the 10053 traces as NewDensity. This will affect cardinality ESTIMATE calculations. The NewDensity is not stored in the data dictionary. The old density values are stored in data dictionary. NewDensity is calculated at run time.

Dynamic sampling (Oracle Database 10g to 11g Change)

Starting with Oracle Database 11g Release 2, the optimizer will automatically decides if dynamic sampling will be useful and what dynamic sampling level will be used for SQL statements executed in parallel. This decision will be based on size of the objects accessed by the statement and the complexity of the predicates. However, if the OPTIMIZER_DYNAMIC_SAMPLING parameter is explicitly set to a non-default value, then that value will honored. If dynamic sampling is being used, it is displayed in the section of the execution plan.

Cardinality Feedback (Oracle Database 10g to 11g Change))

Cardinality feedback is an enhancement made to the Optimizer in Oracle Database 11g Release 2. Cardinality feedback compares cardinality estimates used to derive the plan with the actual cardinality seen in the first execute. If the estimate is 2X off, the cursor is marked for hard parse next time around. The cardinality information seen at first execute is supplied at the next hard parse thus allowing the Optimizer an opportunity to improve on the plan now that it knows more about the actual cardinality seen in the query. If cardinality feedback is used, it is displayed in the section of the execution plan. Cardinality feedback works for predicates on tables, indexes and group by clauses. It does not help for cardinality mis-estimates for joins. Feedback is not persistent on disk, it resides in memory only. The Optimizer will need to "relearn" something if the database is shutdown and restarted. (Related to _optimizer_use_feedback parameter.)

Initialization Parameters

SHARED_POOL_SIZE (Oracle9i to Oracle Database 10g Change)

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 My Oracle Support Document 351018.1: Minimum for SHARED_POOL_SIZE Parameter in 10.2 Version

SESSION_CACHED_CURSORS (Oracle9i to Oracle Database 10g Change)

Prior to Oracle Database 10g, 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 (Oracle9i to Oracle Database 10g Change)

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 (Oracle9i to Oracle Database 10g Change)

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. For assistance with tuning PGA_AGGREGATE_TARGET refer to Oracle Database Performance Tuning Guide 10*g* Release 2>Chapter 7 Memory Configuration and Use

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 (Oracle9i to Oracle Database 10g Change)

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 (Oracle9i to Oracle Database 10g Change)

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 (Oracle 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 NONE. In Oracle Database 10g Release 2, the default value of this parameter is set to SHARED or EXCLUSIVE which have the same meaning.

PARALLEL_ADAPTIVE_MULTI_USER (Oracle9i to Oracle Database 10g Change)

The adaptive multi-user 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 Oracle9i 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 (Oracle9i to Oracle Database 10g Change)

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:

CPU_COUNT x PARALLEL_THREADS_PER_CPU x (2 if PGA_AGGREGATE_TARGET > 0; otherwise 1) x 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 (Oracle9i to Oracle Database 10g Change)

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.

JOB_QUEUE_PROCESSES (Oracle Database 10g to Oracle Database 11g Change)

Beginning with Oracle Database 11g Release 1 (11.1), the JOB_QUEUE_PROCESSES parameter is changed from a basic to a non-basic initialization parameter. Most databases only need to have basic parameters set in order to run properly and efficiently. The default value is also changed from 0 to 1000.

Starting with Oracle Database 11g Release 2 (11.2), setting JOB_QUEUE_PROCESSES to 0 causes both DBMS_SCHEDULER and DBMS_JOB jobs to not run. Previously, setting JOB_QUEUE_PROCESSES to 0 caused DBMS_JOB jobs to not run, but DBMS_SCHEDULER jobs were unaffected and would still run. This parameter will thus affect the running of utlrp.sql after an upgrade to Oracle Database 11g Release 2 (11.2)

Performance and Monitoring

STATISTICS_LEVEL (Oracle9i to Oracle Database 10g Change)

The default STATISTICS_LEVEL is TYPICAL in both Oracle9i, Oracle Database 10g, and Oracle Database 11g. Starting 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 (Oracle9i to Oracle Database 10g Change)

In Oracle9i 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 Database 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 (Oracle9i to Oracle Database 10g Change)

In Oracle Database 10g Release 2, a complete refresh of a single materialized view using dbms_mview.refresh performs a delete of the materialized view base table instead of a truncate (prior to Oracle Database 10g Release 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). Previously, the refresh could 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 (Oracle9i to Oracle Database 10g Change)

'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.

Hang Manager (Oracle Database 10g to 11g Change)

Oracle Database 11g introduces hang manager. This feature is enabled by default. The hang manager automatically detects, analyzes, and dumps diagnostic information for hangs in Oracle Database environments. When a hang is detected the hang manager automatically generates diagnostic trace files. The background process DIA0 (diagnosability process 0) is responsible for hang detection and deadlock resolution. This feature will help you to more quickly resolve database hangs.

Wait Events Statistics (Oracle Database 10g to 11g Change)

Oracle Database 11g Release 1 introduces a change in the way wait counts and time outs are accumulated for wait events (for example, in the V $SYSTEM_EVENT view).

Within Oracle, continuous waits for certain types of resources (such as enqueues) are internally broken into a set of shorter wait calls. Prior to Oracle Database 11g, each individual internal wait call was counted as a separate wait. Starting with Oracle Database 11g Release 1, a single resource wait is recorded as a single wait, irrespective of the number of internal time outs experienced by the session during the wait.

This change allows Oracle to display a more representative wait count, and an accurate total time spent waiting for the resource. Time outs now refer to the resource wait instead of the individual internal wait calls.

As the way waits and time outs are measured has changed, this also affects the average wait time, and the maximum wait time.

For example, there is a problem in the database that a user session must wait for an enqueue it requires (for example, a TX row lock to update a single row in a table) before it can continue processing. Assume it takes 10 seconds to acquire the enqueue. From the sessions perspective, it begins the enqueue wait. Internally to Oracle, this enqueue wait is broken down into 3-second wait calls. In this case, there will be three 3-second wait calls, followed by one 1-second wait call.

In prior releases, the V$SYSTEM_EVENT view would represent this wait scenario as follows:

- TOTAL_WAITS: 4 waits (three 3-second waits, one 1-second wait)
- TOTAL_TIMEOUTS: 3 time outs (the first three waits time out and the enqueue is acquired during the final wait)
- TIME_WAITED: 10 seconds (sum of the times from the 4 waits)
- AVERAGE_WAIT: 2.5 seconds
- MAX_WAIT: 3 seconds

In Oracle Database 11g, this wait scenario is represented as:

- TOTAL_WAITS: 1 wait (one 10-second wait)
- TOTAL_TIMEOUTS: 0 time outs (the enqueue is acquired during the resource wait)
- TIME_WAITED: 10 seconds (time for the resource wait)
- AVERAGE_WAIT: 10 seconds
- MAX_WAIT: 10 seconds

The following common wait events are affected by this change:

Enqueue waits (for example, enq: <name> - <reason> waits)

- Library cache lock waits
- Library cache pin waits
- Row cache lock waits

Statistics affected are listed below:

- Wait counts
- Wait time outs
- Average wait time
- Maximum wait time

Automatic SQL Tuning Advisor (Oracle Database 10g to 11g Change)

This is a new feature in Oracle Database 11g and it is enabled on by default. In Oracle Database 10g, the DBA manually invoked SQL Tuning Advisor to tune SQL. In Oracle Database 11g, Oracle Database automatically runs the SQL Tuning Advisor on selected high-load SQL statements from the Automatic Workload Repository (AWR) that qualify as tuning candidates. This task, called Automatic SQL Tuning, runs in the default Maintenance windows on a nightly basis. You can customize attributes of the maintenance windows, including start and end time, frequency, and days of the week. This AUTO SQL TUNE Task may take resources during the maintenance windows in the process of trying to tune the query because it does actual execution of the query to be tuned.

To disable automatic SQL tuning, use the DISABLE procedure in the DBMS_AUTO_TASK_ADMIN package:

```
BEGIN
DBMS_AUTO_TASK_ADMIN.DISABLE(
client_name => 'sql tuning advisor',
operation => NULL,
window_name => NULL);
END;
```

To enable automatic SQL tuning, use the ENABLE procedure in the DBMS_AUTO_TASK_ADMIN package:

```
BEGIN
DBMS_AUTO_TASK_ADMIN.ENABLE(
client_name => 'sql tuning advisor',
operation => NULL,
window_name => NULL);
END;
```

Administration

CONNECT Role Privileges (Oracle9i to Oracle Database 10g Change)

In Oracle Database 10g Release 2 the CONNECT role only contains the CREATE SESSION privilege. This change enforces good security practices. 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, will now fail due to insufficient privileges. To avoid the failure, grant the specific required privileges prior to upgrading to the user or role.

FAILED_LOGIN_ATTEMPTS (Oracle9i to Oracle Database 10g Change)

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

Prior to Oracle Database 10g Release 2, the default was UNLIMITED.

PASSWORD_GRACE_TIME (Oracle10g to Oracle Database 11g Change)

As of Oracle Database 11g Release 1, the limit for PASSWORD_GRACE_TIME for the DEFAULT profile is 7.

Prior to Oracle Database 11g Release 1, the default was UNLIMITED.

PASSWORD_LIFE_TIME (Oracle10g to Oracle Database 11g Change)

As of Oracle Database 11g Release 1, the limit for PASSWORD_LIFE_TIME for the DEFAULT profile is 180.

Prior to Oracle Database 11g Release 1, the default was UNLIMITED.

PASSWORD_LOCK_TIME (Oracle10g to Oracle Database 11g Change)

As of Oracle Database 11g Release 1, the limit for PASSWORD_LOCK_TIME for the DEFAULT profile is 1.

Prior to Oracle Database 11g Release 1, the default was UNLIMITED.

Change in location of alert.log and other trace files (Oracle Database 10g to 11g Change)

In earlier version the alert.log is located in background_dump_dest. In Oracle Database 11g the alert.log is stored in a new location called as Automatic Diagnostic Repository (ADR)

To locate alert.log

cd <DIAGNOSTIC_DEST>/diag/rdbms/db_name/instance_name/trace

As of Oracle Database 11g Release 1, the diagnostics for each database instance are located in a dedicated directory, which can be specified through the DIAGNOSTIC_DEST initialization parameter. The initialization parameters BACKGROUND_DUMP_DEST and USER_DUMP_DEST are deprecated. They are replaced by the initialization parameter DIAGNOSTIC_DEST, which identifies the location of the ADR.

The structure of the directory specified by DIAGNOSTIC_DEST is as follows:
<*diagnostic_dest*>/diag/rdbms/<*dbname*>/<*instname*>

This location is known as the Automatic Diagnostic Repository (ADR) Home. For example, if the database name is proddb and the instance name is proddb1, the ADR home directory would be <*diagnostic_dest*>/diag/rdbms/proddb/proddb1. The following files are located under the ADR home directory:

- Trace files - located in subdirectory <*adr-home*>/trace
- Alert logs - located in subdirectory <*adr-home*>/alert. In addition, the alert. log file is now in XML format, which conforms to the Oracle ARB logging standard.

The alert log is an XML file that is a chronological log of database messages and

errors. You can view the alert log in text format (with the XML tags stripped) with Enterprise Manager and with the ADRCI utility. There is also a text-formatted version of the alert log stored in the ADR for backward compatibility. However, Oracle recommends that any parsing of the alert log contents be done with the XML-formatted version, because the text format is unstructured and may change from release to release.

- Core files - located in the subdirectory <*adr-home*>/cdumd
- Incident files - the occurrence of each serious error (for example, ORA-600, ORA-1578, ORA-7445) causes an incident to be created. Each incident is assigned an ID and dumping for each incident (error stack, call stack, block dumps, and so on) is stored in its own file, separated from process trace files. Incident dump files are located in <*adr home*>/incident/<*incdir#*>. You can find the incident dump file location inside the process trace file.

New .trm files.

In Oracle Database 11g trace files (.trc), sometimes are accompanied by corresponding trace map (.trm) files, which contain structural information about trace files and are used for searching and navigation. The .trm files are small in size. So you can expect to see more files generated as compared to earlier releases, but do not get alarmed by many .trm files as they are expected with the . trc files.

For in-depth details on ADR see the Oracle Database documentation set.

Oracle Database Administrator's Guide - Automatic Diagnostic Repository (ADR)

New Default Value for UNDO_MANAGEMENT (Oracle Database 10g to 11g Change)

Starting with Oracle Database 11g Release 1, the default value of the UNDO_MANAGEMENT parameter is AUTO so that automatic undo management is enabled by default. You must set the parameter to MANUAL to turn off automatic undo management, required. In Oracle Database 10g and Oracle9i , the UNDO_MANAGEMENT parameter is set to MANUAL by default. A PL/SQL procedure DBMS_UNDO_ADV.RBU_MIGRATION is provided to help properly size the Undo tablespace for each individual environment. This feature facilitates seamless migration to AUM from databases being upgraded to the new release so that they can start taking advantage of all the benefits of AUM.

Auto-Task (Oracle Database 10g to 11g Change)

In Oracle Database 11g the automated maintenance tasks infrastructure known as AutoTask enables Oracle Database to automatically schedule Automatic Maintenance Tasks. AutoTask schedules automatic maintenance tasks to run in a set of Oracle Scheduler windows known as maintenance windows. Maintenance windows are those windows that are members of the Oracle Scheduler window group MAINTENANCE_WINDOW_GROUP.

The following are the tasks that AutoTask automatically schedules in these maintenance windows:

- Optimizer statistics gathering
- Automatic Segment Advisor
- SQL Tuning Advisor

By default there are seven predefined maintenance windows, each one representing a day of the week. The weekend maintenance windows, SATURDAY_WINDOW and SUNDAY_WINDOW, are longer in duration than the weekday maintenance windows. The window group MAINTENANCE_WINDOW_GROUP consists of these seven windows.

Window Name Description	
MONDAY_WINDOW	Starts at 10 p.m. on Monday and ends at 2 a.m.
TUESDAY_WINDOW	Starts at 10 p.m. on Tuesday and ends at 2 a.m.
WEDNESDAY_WINDOW	Starts at 10 p.m. on Wednesday and ends at 2 a.m.
THURSDAY_WINDOW	Starts at 10 p.m. on Thursday and ends at 2 a.m.
FRIDAY_WINDOW	Starts at 10 p.m. on Friday and ends at 2 a.m.
SATURDAY_WINDOW	Starts at 6 a.m. on Saturday and is 20 hours long
SUNDAY_WINDOW	Starts at 6 a.m. on Sunday and is 20 hours long.

Auto-Task provides the necessary infrastructure to handle large numbers of jobs. Auto-Task provides out-of-the-box management of resource distribution (CPU and I/O) among the various database maintenance tasks such as Automatic Optimizer Statistics Collection, Automatic Segment Advisor, and others. The CPU is automatically managed. I/O is managed only if the I/O Resource Manager is enabled.

Auto-Task ensures that work during maintenance operations is not affected and that user activity gets the necessary resources to complete.

You can disable a particular automated maintenance tasks for all maintenance windows with a single operation. You do so by calling the DISABLE procedure of the DBMS_AUTO_TASK_ADMIN PL/SQL package without supplying the window_name argument. For example, you can completely disable the Automatic SQL Tuning Advisor task as follows:

```
BEGIN
dbms_auto_task_admin.disable(
client_name => 'sql tuning advisor',
operation => NULL,
window_name => NULL);
```

```
END;
```

To enable this maintenance task again, use the ENABLE procedure, as follows:

```
BEGIN
dbms_auto_task_admin.enable(
client_name => 'sql tuning advisor',
operation => NULL,
window_name => NULL);
END;
```

The task names to use for the client_name argument are listed in the DBA_AUTOTASK_CLIENT database dictionary view.

To enable or disable all automated maintenance tasks for all windows, call the ENABLE or DISABLE procedure with no arguments.

```
EXECUTE DBMS_AUTO_TASK_ADMIN.ENABLE;
EXECUTE DBMS_AUTO_TASK_ADMIN.DISABLE;
```

Automatic Maintenance Tasks Management is automatically enabled when upgrading to Oracle Database 11g. If you disable either Optimizer Statistics Gathering or Segment Advisor jobs in Oracle Database 10g, then the corresponding Automatic Maintenance Tasks Management feature is disabled after upgrading to Oracle Database 11g.

TIMESTAMP WITH TIME ZONE data (Oracle Database 11gR1 to 11gR2 Change)

When time zone file is upgraded to a new version due to Daylight Saving Time changes, TIMESTAMP WITH TIMEZONE (TSTZ) data could become stale. In previous releases, database administrators ran the SQL script utltzuv2.sql to detect TSTZ data affected by the time zone version changes and then had to carry out extensive manual procedures to update the TSTZ data.

In Oracle Database 11g Release 2, TSTZ data can be updated transparently with minimal manual procedures and system downtime using the newly provided DBMS_DST PL/SQL packages. For convenience, starting with Oracle Database 11g Release 2, time zone files from version 1 to 14 are automatically installed in the directory of $ORACLE_HOME/ORACORE/ZONEINFO/. After the database has been upgraded, database administrators have the option of staying with the current time zone version or upgrade to the latest. Oracle recommends upgrading the server to the latest timezone version after a database has been upgraded.

Time zone files are also installed on the clients. Starting in Oracle Database 11g Release 2, there is no longer a need for clients to upgrade their time zone files immediately. Upgrades can be done at a time when it is most convenient to the system administrator. However, there could be a small performance penalty when the client and server use different time zone versions.

Starting with Oracle Database 11g Release 2(11.2.0.2), DBUA offers a choice to automatically upgrade "TIMESTAMP WITH TIME ZONE data " and does this upgrade as part of post upgrade steps.

After DBMS_DST has been run, a new time zone file version is used. At this point, in order to downgrade, a time zone file of the same version must be installed in the older oracle home prior to the downgrade.

See also:

Oracle Database Upgrade Guide > Chapter 3 > TIMESTAMP WITH TIME ZONE Data Type information about preparing to upgrade Timestamp with Time Zone data, Oracle Database Globalization Support Guide, Chapter 4 for information about how to upgrade the Time Zone file and Timestamp with Time Zone data, and Oracle Call Interface Programmer's Guide for information about performance affects of clients and servers operating with different versions of Time Zone files.

Datafile Write Errors (Oracle Database 11gR1 to 11gR2 Change)

Starting with the 11.2.0.2 patchset for Oracle Database 11g Release 2, a write error to any data file will cause the database to perform a shutdown abort of the instance.

In prior releases, I/O errors to datafiles not in the system tablespace would offline the respective datafiles when the database is in archivelog mode. This behavior is not always desirable. Some customers would prefer that the instance crash due to a datafile write error. A new hidden parameter called _datafile_write_errors_crash_instance has been introduced to control whether the instance should crash on a write error or if the datafiles should be taken offline on a write error.

- If _datafile_write_errors_crash_instance = TRUE (default) then any write to a datafile which fails due to an IO error causes an instance crash.
- If _datafile_write_errors_crash_instance = FALSE then the behavior reverts to the previous behavior (before this fix) such that a write error to a datafile offlines the file (provided the DB is in archivelog mode and the file is not in SYSTEM tablespace in which case the instance is aborted)

Also see My Oracle Support Document 7691270.8.

Streams

Initialization Parameters (Oracle9i to Oracle Database 10g Change)

1. ***AQ_TM_PROCESSES***

This parameter rules the behavior of Time Monitor background process. The behavior of this parameter has radically changed between Oracle9i Release 2 and Oracle Database 10g Release 2.

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 and above, 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. My Oracle Support Document 428441.1: "Warning: Aq_tm_processes Is Set To 0" Message in Alert Log After Upgrade to 10.2.0.3 or Higher 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 (Oracle9i to Oracle Database 10g Change)

Several architectural changes have been introduced in Oracle Database 10g Release 2 and above, 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 and later releases, 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=200M***
- ***shared_pool_size=0***

Due to this change hidden parameters _kghdsidx_count and _first_spare_parameter are no longer required.

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 manual flow control.

In Oracle Database 10g and later releases, the flow control process is now internalized and automatic and does not require any setup. It will pause the capture to avoid excessive spill over. Manual flow control from earlier releases should be removed.

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 and later releases, purge LogMiner checkpoint metadata is an automatic process controlled by the capture parameter checkpoint_retention_time. It is recommended that the default value (60 days) be reduced to match customer requirements. A typical customer setting for this is 2 (days). Also it is recommended to use the default setting of the LogMiner checkpoints (1000M).

In Oracle Database 10g Release 2 and Oracle Database 11g Release 1, if the capture parameter _checkpoint_frequency is explicitly set to a lower value, reset it to 1000 through the capture parameter _checkpoint_frequency.
In Oracle Database 11g Release 2, If the capture parameter _checkpoint_frequency is explicitly set to a lower value, reset this capture parameter by specifying its value as NULL in the SET__PARAMETER procedure of DBMS_CAPTURE_ADM.

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 and later releases, 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 and later releases 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 Oracle9i Release 2 database. This supplemental logging must be explicitly configured on the source database in Oracle Database 10g Release 2 and later releases.

5. ***Name of processes***

In Oracle9i Release 2:

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

In Oracle Database 10g:

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

6. ***Simplified Apply Handling of LOB columns***

In Oracle9i Database Release 2, multiple logical change records make up a single LOB column and are processed individually by an apply handler.

In Oracle Database 10g Release 2 and later releases, the entire LOB column can be processed as a single column by setting the ASSEMBLE_LOBS parameter to TRUE with a registered apply handler (SET_DML_HANDLER procedure of DBMS_APPLY_ADM package).

7. ***Oracle Streams and Oracle Real Application Clusters (Oracle RAC)***

In Oracle 9i Release 2, queue ownership migrated automatically to a surviving instance of an Oracle RAC database. Associated Streams processes had to be manually restarted after the migration of the queue ownership. In addition, Streams capture on an Oracle RAC database processed only from the archive logs of the database. If an instance was unavailable, the Streams capture process waited until all log threads (including the unavailable threads) were archived before processing. To workaround this delay in processing, a job was created to force the archival of all closed threads so that capture could continue processing. Propagation to an Oracle RAC database had to be directed to the owning instance of the receiving queue at the target database.

In Oracle Database 10g Release 2 and later releases, the Streams processes automatically restart after queue ownership is migrated to a surviving instance. Specific instances can be specified as the primary and secondary instances for queue ownership using the ALTER_QUEUE_TABLE procedure of DBMS_AQADM. In addition, Streams capture will continue processing even if an instance thread is unavailable. The Streams capture process mines the online redo logs of each thread as well as from the archived redo logs.

In Oracle Database 10g Release 2 and later releases, queues created using set_up_queue of the DBMS_STREAMS_ADM package on an Oracle RAC database automatically create a service name for the queue. This service name is invoked automatically on an Oracle RAC database when a propagation is created with the queue_to_queue parameter set to TRUE.

Architectural Changes (Oracle Database 10g to 11g Change)

Several architectural changes have been introduced in Oracle Database 11g compared to release Oracle Database 10g.

1. ***Apply Handler Modification Necessary for Handling New Error Messages that Replace ORA-1403 Errors***

In Oracle Database 11g, customized DML and error handlers for Streams require modification to catch the additional Oracle errors ORA-26786 and ORA-26787 in place of the ORA-01403 No data found message.

An ORA-26787 error is raised if the row to be updated or deleted does not exist in the target table.

An ORA-26786 error is raised when the row exists in the target table, but the values of some columns do not match those of the LCR.

2. ***Name of processes***

In Oracle Database 10g:

- ***Capture processes are called Cxxx and may use as many Parallel slave processes as required for capture parallelism.***
- ***Apply processes are called Axxx and may use as many Parallel slave processes as required for apply parallelism.***

In Oracle Database 11g

- ***Capture processes are called CPxx, with sub-processes named MSxx. No parallel slave processes are required for capture parallelism.***
- ***Apply processes are called APxx, with sub-processes named ASxx. No parallel slave processes are required for apply parallelism.***

3. ***Event monitor (EMON) name change***

In Oracle9i Release 2 and Oracle Database 10g, the event monitor process (EMON) is a single process within the database.

In Oracle Database 11g, the event monitor process is now a set of processes consisting of a coordinator process (EMON) and four (4) subprocesses (E000, E001, E002, E003) on each instance of the database. Notification for events is performed at the originating instance of the event.

4. Propagation performed via Database Scheduler

The propagation job is created from the DBMS_PROPAGATION_ADM package. In Oracle9i Release 2 and Oracle Database 10g, the job is created as part of DBMS_JOB. In Oracle Database 11g, it is created as a job via DBMS_SCHEDULER. The job creation is transparent - but for monitoring the job, you need to know which job scheduler is used.

In Oracle9i Release 2 and Oracle Database 10g, Oracle Streams propagation jobs were performed via the JOB system (DBMS_JOB package, DBA_JOBS view).

Starting in Oracle Database 11g, Oracle Streams propagation jobs are performed via the Oracle Scheduler (DBMS_SCHEDULER package, DBA_SCHEDULER_JOBS view). Streams propagation jobs are managed using the DBMS_PROPAGATION_ADM package procedures start_propagation and stop_propagation.

After upgrading to Oracle Database 11g, Oracle Streams Advanced Queuing propagation jobs must be managed with the appropriate procedures from the DBMS_AQADM package (enable_propagation_schedule, disable_propagation_schedule, schedule_propagation, unschedule_propagation).

Security

Password related changes (Oracle Database 10g to 11g Change)

In Oracle Database 11g password are case sensitive. In previous versions the passwords are not case sensitive. This feature is enabled by the initialization parameter sec_case_sensitive_logon that is TRUE by default. Setting sec_case_sensitive_logon to FALSE disables the case sensitive feature. After migrating the Oracle database from previous versions the existing user passwords are case-insensitive until users change it. A new column PASSWORD_VERSIONS is added to the dba_users view. PASSWORD_VERSIONS Column shows The Database version in which the password was created or changed.

```
SELECT USERNAME, PASSWORD_VERSIONS FROM DBA_USERS;

 USERNAME                       PASSWORD_VERSIONS
 ------------------------------ -----------------
 JONES                          10G 11G
 ADAMS                          10G 11G
 CLARK                          10G 11G
 PRESTON                        11G
 BLAKE                          10G
```

The passwords for accounts jones, adams, and clark were originally created in Oracle Database 10g and then reset in Oracle Database 11g. Their passwords, assuming case sensitivity has been enabled, are now case sensitive, as is the password for preston. However, the account for blake is still using the Oracle Database 10g standard, so it is case insensitive.

You can enable or disable case sensitivity for password files by using the ignorecase argument in the ORAPWD command line utility. The default value for ignorecase is n (no), which enforces case sensitivity.

DEFAULT profile change (Oracle Database 10g to 11g Change)

The DEFAULT profile in Oracle Database 11g has been changed.

The values of PASSWORD_GRACE_TIME, PASSWORD_LIFE_TIME, PASSWORD_LOCK_TIME have been changed from UNLIMITED to 7,180,1 respectively.

RESOURCE_NAME	10.2.X.0	11.1.0.X
PASSWORD_GRACE_TIME	UNLIMITED	7
PASSWORD_LIFE_TIME	UNLIMITED	180
PASSWORD_LOCK_TIME	UNLIMITED	1

The upgrade from earlier versions to Oracle Database 11g will not change the existing profile settings, but any new user created in Oracle Database 11g with default profile settings will be impacted.

Auditing turned on by default (Oracle Database 10g to 11g Change)

In Oracle Database 11g the parameter AUDIT_TRAIL by default is now set to DB. The auditing is turned on by default in Oracle Database 11g. In earlier versions auditing was not turned on by default. The AUDIT_TRAIL parameter by default was false.

Oracle Database audits the AUDIT ROLE SQL statement by default.

The privileges that are audited by default are as follows:

```
ALTER ANY PROCEDURE
CREATE ANY LIBRARY
DROP ANY TABLE
ALTER ANY TABLE
CREATE ANY PROCEDURE
DROP PROFILE
ALTER DATABASE
CREATE ANY TABLE
DROP USER
ALTER PROFILE
CREATE EXTERNAL JOB
EXEMPT ACCESS POLICY
ALTER SYSTEM
CREATE PUBLIC DB LINK
GRANT ANY OBJECT PRIVILEGE
ALTER USER
CREATE SESSION
GRANT ANY PRIVILEGE
AUDIT SYSTEM
CREATE USER
```

```
GRANT ANY ROLE
CREATE ANY JOB
DROP ANY PROCEDURE
```

Oracle Database also audits all privileges and statements that have the BY ACCESS clause. This information by default is written to SYS.AUD$ table. There is some performance overhead for auditing. In addition, the database administrator will need to implement a purge policy for the SYS.AUD$ table using DBMS_AUDIT_MGMT pl/sql package. The DBMS_AUDIT_MGMT pl/sql package was made available in Oracle Database 11g Release 2. Setting AUDIT_TRAIL=NONE disables the default standard auditing. The auditing is turned on by default only if you choose the default secure configuration option settings during DBCA database creation of a new Oracle 11g Database. For existing pre Oracle 11g databases, this auditing option is not turned on by default due to upgrade.

Fine-Grained Access to Database Network Services. (Oracle Database 10g to 11g Change)

A new security measure is introduced in this release for the following network-related PL/SQL packages: UTL_TCP, UTL_HTTP, UTL_SMTP, UTL_MAIL, and UTL_INADDR. The invoker of those packages needs additional privileges to connect to an external host or to resolve the name or the IP address of a host. The packages check the invoker for the necessary privileges only when the calls are made at runtime and raises an exception if the invoker lacks the privileges. This new security measure is implemented by the XML DB access control list (ACL) mechanism and, therefore, requires XML DB to be installed in order to use those packages. This New feature, which is enabled by default, enhances security for network connections because it restricts the external network hosts that a database user can connect to using the PL/SQL network utility packages such as UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, and UTL_INADDR. Otherwise, an intruder who gained access to the database could maliciously attack the network, because, by default, the PL/SQL utility packages are created with the EXECUTE privilege granted to PUBLIC users.

If you have upgraded from a previous release of Oracle Database, and your applications depend on PL/SQL network utility packages such as UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, and UTL_INADDR, the following error may occur when you try to run the application:

```
ORA-24247: network access denied by access control list (ACL)

   Cause:    No access control list (ACL) has been assigned to the target
             host or the privilege necessary to access the target host has not
             been granted to the user in the access control list.
   Action:   Ensure that an access control list (ACL) has been assigned to
             the target host and the privilege necessary to access the target
                          host has been granted to the user.
```

These packages do not allow connections to external network services to a non privileged users by default. Earlier version allow to connections to external network services to a non privileged users by default. In Oracle Database 11g you can use the DBMS_NETWORK_ACL_ADMIN package to create and manage access control list.

DBMS_SQL Package (Oracle Database 10g to 11g Change)

In Oracle Database 11g Release 1, Oracle introduces a number of enhancements to DBMS_SQL to improve the security of the package:

Prevent guessing of open cursor numbers

A new error, ORA-29471, will be raised when any DBMS_SQL subprogram is called with a cursor number that does not denote an open cursor. When the error is raised, an alert is issued to the alert log and DBMS_SQL becomes inoperable for the life of the session. If the actual value for the cursor number in a call to IS_OPEN does denote a cursor that is currently open in the session, the return value is TRUE. If the actual is null, the return value is FALSE. Otherwise, you get the ORA-29471 error. Note that the DBMS_SQL.OPEN_CURSOR function is the only DBMS_SQL subprogram that has no formal parameter for the cursor number. Rather, it returns a cursor number. Therefore, it is not within the scope of the rules.

Prevent inappropriate use of a cursor

Cursors are now better protected from security breaches that subvert known, existing cursors. Checks are always made when binding and executing. Optionally, checks may be performed for every single DBMS_SQL subprogram call. The check is :current_user is the same on calling the subprogram in question as it was on calling the most recent parse. The enabled roles on calling the subprogram must be a superset of the enabled roles on calling the most recent parse. As is always the case, for definer's right subprograms, roles are irrelevant. If either check fails, ORA-29470 is raised.

The mechanism for defining when checks are performed is a new overload for the OPEN_CURSOR subprogram which takes a formal parameter, security_level, with allowed values NULL, 1 and 2.

When security_level = 1 (or is NULL), the checks are made only when binding and executing.

When security_level = 2, the checks are always made.

security_level is a parameter is to be set with DBMS_SQL.OPEN_CURSOR

syntax is:

```
DBMS_SQL.OPEN_CURSOR (
security_level IN INTEGER)
RETURN INTEGER;
```

This security regime is stricter than in Oracle Database 10g Release 2 and previous releases. As a consequence, users of DBMS_SQL may encounter runtime errors on upgrade. While the regime makes for more secure applications, users may want to relax the security checks temporarily as they migrate to Oracle Database 11g. If so, please consult with Oracle Support Services on steps to relax the security checks.

SEC_MAX_FAILED_LOGIN_ATTEMPTS (Oracle Database 10g to 11g Change)

This is a new initialization parameter in Oracle Database 11g. The default value is 10.

SEC_MAX_FAILED_LOGIN_ATTEMPTS specifies the number of authentication attempts that can be made by a client on a connection to the server process. After the specified number of failure attempts, the connection will be automatically dropped by the server process.

With Oracle Database, a server process is first started, and then the client authenticates with this server process. An intruder could start a server process first, and then issue an unlimited number of authenticated requests with different user names and passwords in an attempt to gain access to the database.

You can limit the number of failed login attempts for application connections by setting the SEC_MAX_FAILED_LOGIN_ATTEMPTS initialization parameter to restrict the number of authentication attempts on a connection. After the specified number of authentication attempts fail, the database process drops the connection. By default, SEC_MAX_FAILED_LOGIN_ATTEMPTS is set to 10.

Remember that the SEC_MAX_FAILED_LOGIN_ATTEMPTS initialization parameter is designed to prevent potential intruders from attacking your applications; it does not apply to valid users. The sqlnet.ora INBOUND_CONNECT_TIMEOUT parameter and the FAILED_LOGIN_ATTEMPTS initialization parameter also restrict failed logins, but the difference is that these two parameters only apply to valid user accounts.

For example, to limit the maximum attempts to 5, set SEC_MAX_FAILED_LOGIN_ATTEMPTS as follows in the initsid.ora initialization parameter file:

SEC_MAX_FAILED_LOGIN_ATTEMPTS = 5

RAC/ASM

Oracle Database 11g Release 2 introduces the Oracle Grid Infrastructure. For clusters, we have combined the binaries for Oracle Clusterware and Automatic Storage Management to a single home, which is referred to as the grid infrastructure home (grid_home). When you upgrade to Oracle Database 11g Release 2, you will have to upgrade both ASM and Oracle Clusterware at the same time. It is not supported to run Oracle Clusterware 11g Release 2 with an earlier release of Automatic Storage Management (ASM).

For standalone servers running single instance Oracle database with ASM, you will need to install the Grid Infrastructure for a Standalone Server. This home will contain Automatic Storage Management and a new feature called Oracle Restart. Oracle Restart includes the Oracle High Availability Services daemon, which will automatically start your Oracle resources (IE database, ASM, listener) when the server starts, monitor the resources and restart them if they fail, and stop them with the server is shutdown. To install the Grid Infrastructure for a standalone server , you must start the installer from the clusterware directory on the distribution media. (Oracle Internal Note: This directory should change to grid from clusterware by product release).

The Grid Infrastructure installation includes a software only installation; this option is

an advanced option and only recommended for experienced Oracle RAC customers who require advanced deployment options.

The Grid Infrastructure for a cluster installation offers 2 options consistent with the Oracle Database Installation. The simplest install method is to use the "typical install" which streamlines the install and requires a minimal amount of input from the user. This install assumes you will take advantage of the new features Grid Plug and Play and the OCR and Voting Disk will be stored in ASM. Before starting this install, you will need to follow the pre-requisites in the Oracle Grid Infrastructure for a Cluster Installation Guide for your platform. The second method of install is the "Advanced Installation", it requires more steps however it allows you to customize the install.

NOTE: If you have not already done so, it is strongly recommended that you review My Oracle Support Document 1363369.1 Things to Consider Before Upgrading to 11.2.0.3 Grid Infrastructure/ASM, Document 1212703.1: Oracle Grid Infrastructure 11.2.0.2 Installation or Upgrade may fail due to Multicasting Requirement and Document 1210883.1: 11gR2 Grid Infrastructure Redundant Interconnect and ora.cluster_interconnect.haip.

CONTROL_MANAGEMENT_PACK_ACCESS (Oracle Database 10g to 11g Change)

In Oracle Database 10g, ADDM was enabled by default. In order to disable ADDM, you would have followed the instructions in My Oracle Support Document 562932.1: How To Disable Addm . With the release of Oracle Database 11g, a new parameter called control_management_pack_access specifies which of the Server Manageability Packs should be active. The default value is "DIAGNOSTIC +TUNING". To disable ADDM, you would set control_management_pack_access to "NONE".

Integration of Oracle Universal Installer and the Cluster Verification Utility (Oracle Database 10g to 11g Change)

The Installer uses the cluster verification utility to complete all prerequisite checks during the interview process. You no longer have to manually execute all the CVU commands prior to starting your install. The installer will set up the required SSH user equivalence. If prerequisite checks fail, you will be prompted with detailed explanation of what the problem is and how to fix it. Where possible, a fixup script is generated. It is recommended that you execute the fixup script as directed to complete all prerequisites before continuing with the install.

Managing Your Oracle RAC Databases (Oracle Database 10g to 11g Change)

Oracle RAC 11g introduces a new way to manage your Oracle RAC databases within a cluster. Traditionally we have allocated specific Oracle RAC instances to nodes within the cluster that requires the DBA to hard code parameters such as instance number, redo thread etc. If that node in the cluster did not start, then the instance for the database does not start. This traditional method of managing your Oracle RAC database will continue to be available and will be referred to as Administrator Managed database (Admin-Managed). Any database upgraded from an earlier release of Oracle Database, will be Administrator Managed. Any database that is using an earlier version of Oracle Database will be Administrator Managed. These databases will be managed using the same commands or methods (such as DBCA or Enterprise Manager) as they have been with previous releases of Oracle Database. All commands and utilities have maintained backward compatibility.

To move away from any hard coding of parameters, to make it easier to replace nodes in a cluster or expand the cluster as requirements change, we introduce a new way of managing your Oracle RAC database called Policy Managed. Policy Managed databases must be 11g Release 2 and cannot co-exist on the same servers as administrator managed databases. A policy managed database is defined by cardinality, the number of instances you would like running during normal operations. A policy-managed database will run in one or more database server pools that are created in the cluster by the cluster administrator. An instance will start on all servers that are in the server pools defined for the database. If you are using Automatic Storage Management (ASM) for your database storage, when an instance starts, if there is not a redo thread available, one will automatically be enabled and the required redo log files created as well as the undo tablespace.

With Oracle RAC 11g Release 2, there is only a database resource define to Oracle Clusterware. This resource will contain the Oracle Home, the spfile, the server pool(s) and the ASM diskgroup(s) required for the database. The database resource will have a weak start dependency on the VIP which means it will try to start the VIP for the node when the instance starts however if the VIP does not start successfully, the instance will still be started. When reviewing the database resource for an Administrator Managed database, you will see a server pool defined with the same name as the Oracle database. This pool will be part of a special Oracle defined server pool called Generic. The Generic server pool is managed by Oracle to support Administrator Managed databases. The server pools that are part of Generic will be automatically created or removed when you add or remove an Administrator Managed database.

Services for Administrator Managed databases will continue to be defined by the Preferred and Available definitions. For Policy Managed Databases, a Service will be defined to a Database Server Pool and can either be uniform (running on all instances in the server pool) or Singleton (running on only 1 instance in the server pool).

For more information on Managing Service refer to Oracle Real Application Clusters Administration and Deployment Guide 11g: Introduction to Automatic Workload Management.

Cluster Managed Services (Oracle Database 10g to 11g Change)

The database service attributes (those defined by the dbms_service pl/sql package) are now stored in the Oracle Clusterware resource for the service. All modifications to cluster managed services must be done by either srvctl or Enterprise Manager. Whenever the cluster starts a service, it verifies that the definition in the Oracle Clusterware (CRS) resource is reflected in the database, if there is a difference, it will modify the database to match the CRS resource.

Cluster Startup with Oracle Clusterware (Oracle Database 10g to 11g Change)

Oracle Clusterware (on Unix/Linux) is now started through a single entry in init. This starts the OHASD (Oracle High Availability Services Daemon), which is responsible for starting the rest if the background processes. New background processes with 11g Release 2 (defined in the Oracle Clusterware Admin and Deployment Guide) are:

- OHASD
- MDNSD
- GPNPD
- CTSSD
- GNS
- EONSD

<<<<Pictures go here (Oracle Clusterware TrailBlazer Processes and Startup) >>>>

CRSCTL, the command line interface for managing the cluster, introduces a clusterized start or stop of the cluster. From any node in the cluster, you can enter crsctl stop cluster -all and the clusterware will be stopped on all nodes. Note: Not all background processes will be stopped, you will still see OHASD, MDNSD, and GPNPD running. To start the cluster, enter crsctl start cluster -all and the clusterware will be started on all nodes. This is achieved through the OHASD. If you want all Oracle Clusterware processes stopped, you can use the command *crsctl stop crs*.

ASM (Oracle Database 10g to 11g Change)

ASM diskgroups have 3 compatible attributes. It is important to understand these attributes and their settings. If the attribute is set to a lower release or the attribute is not set, you will not be able to take advantage of new features. For example if you would like to create a file system in ASM (using ASM cluster file system), the compatible.advm attribute for the diskgroup must be set to 11.2.0.0.0. For more information see the Storage Administrators Guide.

The Oracle Cluster Registry (OCR) and Voting disks can now be stored in ASM. The upgrade to Oracle Database 11g Release 2 will not change the storage location for any files (OCR, voting disk or database files). Once you have completed the upgrade, if you want to move your OCR and voting disks to ASM, you can do this using the ocrconfig and crsctl commands as documented in the Oracle Clusterware Admin and Deployment Guide.

The SPFILE for the ASM instance can now be stored in ASM. As above, the upgrade to Oracle Database 11g Release 2 will not change the storage location for any files. To move your SPFILE for ASM to ASM, follow the procedure documented in Storage Administrators Guide.

ASM introduces a new general purpose Volume Manager and Cluster File System. The file system is stored on standard ASM diskgroups. This file system can be used to store your Oracle Database Home. Oracle Databases should continue to use ASM files and not use an ASM Cluster File System.

Connecting to Your Oracle RAC Database (Oracle Database 10g to 11g Change)

Oracle RAC 11g Release 2 introduces the Single Client Access Name (SCAN). SCAN allows you to use a single name to access your RAC database. The client connect strings from your Oracle RAC database prior to upgrade will continue to work without changes. It is recommended that you migrate your client connections to use SCAN after the upgrade is complete. Using SCAN allows you to use simplified connection strings such as the JDBC thin URL or EZConnect. When you use SCAN to connect, you do not have to change the client connect string if you add or remove nodes from the cluster.

NOTE: Applications which have earlier clients (11.1 or earlier) are NOT SCAN aware and listing the SCAN name in tnsnames.ora or other name resolution method will not resolve properly to the three SCAN addresses. Refer to My Oracle Support Document 1058646.1: How to integrate a 10g/11gR1 RAC database with 11gR2 clusterware (SCAN) for details.

For more information on SCAN see the Oracle Real Application Clusters Administration and Deployment Guide.

Grid Plug and Play (Oracle Database 10g to 11g Change)

Grid Plug and Play (GPnP) simplifies addition and removal of nodes from a cluster, making it easier to deploy clusters in a dynamically provisioned environment. Grid Plug and Play also enables databases and services to be managed in a location-independent manner.

GPnP introduces the Grid Naming Service (GNS) and the use of DHCP in the cluster. When GNS and DHCP are enabled in your cluster, all the Virtual IP addresses are dynamically allocated when the cluster is started or a node joins the cluster. To set up GNS, your Network Administrator must forward a domain to an IP address within the corporate DNS. This domain name and IP address are input during the install or Oracle Clusterware. Once the cluster is up the GNS will run on one node in the cluster and provide the name translation for any connection request in the cluster. This eases the cluster administration as once the Network Administrator has set up the forwarded domain, the cluster administrator can add and remove nodes in the cluster as requirements change.

For more information, see the Oracle Clusterware Administration and Deployment Guide.

Patching and Upgrade

Patching (Oracle Database 10g to 11g Change)

Starting with the first patch set for Oracle Database 11g Release 2 (11.2.0.2), Oracle Database patch sets are full installations of the Oracle Database software. See My Oracle Support Document 1189783.1: Important Changes to Oracle Database Patch Sets Starting With 11.2.0.2 for more details.

Out of Place Upgrade for Oracle(Oracle Database 11gR1 to 11gR2 Change)

Oracle recommends that you perform an out-of-place patch set upgrade. It requires much less downtime and is safer as it does not require patching an ORACLE_HOME that is already being used in production. For an out-of-place patch set upgrade, install the patch set into a new, separate Oracle home location. After you install the patch upgrade, you then migrate the Oracle Database from the older Oracle home. The patch set upgrade is now the same process to upgrade from one version to another.

Patches Recommended

This page will help you prepare for an upgrade by pointing to a list of patches that are recommended for your platform. Oracle introduced a set of Recommended Patches which make it easier to obtain and deploy fixes for known critical issues encountered in targeted environments and configurations.

To locate the Recommended Patches do one of the following:

- Use the My Oracle Support Patches & Updates page. See the *Operating System* section below for details. This approach will provide the most current list of Recommended Patches.
- Use My Oracle Support Documents 756388.1 and 756671.1.
  Document 756388.1: Introduction to Oracle Recommended Patches provides an introduction to Oracle Recommended Patches
  Document 756671.1: Oracle Recommended Patches -- Oracle Database provides links to the Recommended Patches which are listed by target configurations.

Carefully review each patch and only apply the patches specific to your environment and the features being used.

Operating System Patches

For convenience, direct links to the recommended patch list on My Oracle Support have been provided for some of the common operating systems. You will need to be logged into My Oracle Support. If you are upgrading on an operating system which is not listed, follow the steps provided for obtaining the list of recommended patches.

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

Field	Value
Product	Oracle Database
Release	Choose the Upgrade Release (i.e Oracle 11.2.0.X)
Platform	Choose a platform

5. Click "Go"

A list of recommended patches will display.

Current Database Patch Sets Schedule

My Oracle Support Document 742060.1: Release Schedule of Current Database Releases provides a schedule and pertinent support information for currently released Oracle Database patch sets. The note represents the most accurate information Oracle can provide, and is updated frequently.

Documentation

Oracle Database Upgrade Guide

This guide is intended for database administrators (DBAs), application developers, security administrators, system operators, and anyone who plans or executes Oracle Database upgrades. The 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 document provides information about how to assure the integrity of database changes using Oracle Real Application Testing. This document is intended for database administrators, application designers, and programmers who are responsible for performing real application testing on Oracle Database.

Oracle Database Real Application Testing User's Guide 11g

This document provides information about how to assure the integrity of database changes using Oracle Real Application Testing. This document is intended for database administrators, application designers, and programmers who are responsible for performing real application testing on Oracle Database.

Oracle Database Readme

This Readme file is relevant only to the delivered Oracle Database 11g 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 for UNIX Based Operating Systems

This guide is intended for anyone responsible for administering and configuring Oracle Database 11g. If you are configuring Oracle Real Application Clusters (Oracle RAC), then refer to Oracle Clusterware Administration and Deployment Guide and Oracle Real Application Clusters Administration and Deployment Guide.

Oracle Database Installation List

These guides describe how to install and configure Oracle Database 11g. Specifically, review Chapter 1 for Upgrade Recommendations. These guides are intended for anyone responsible for installing Oracle Database 11g.

Related Documentation

Oracle Database 11g 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 - 11g Release 2 (11.2) for Windows and UNIX

This chapter is designed to aid the developers, administrators, and all other users who install Oracle software (Oracle Database 11g Release 1 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 11g 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 Oracle Enterprise Manager Grid Control Advanced Installation and Configuration Guide 11g Release 1 - Upgrading to Enterprise Manager 11g

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 are updated periodically to reflect changes in transition rules for various time zone regions. The files supplied with Oracle Database 11g Release 1 have been updated to version 4; and version 8 for Oracle Database 11g Release 2.

Oracle Database Storage Administrator's Guide

This guide is intended for database and storage administrators who administer and manage storage and configure and administer ASM.