Oracle DBA Code Examples

Version Date: 30 July 2011

Editor: Ahmed Baraka

Document Purpose

This document is edited to be a quick reference of code examples used to achieve specific Oracle DBA tasks. No explanation on any subject is presented.

The document is simply oriented based on the required task, the code to perform the task and any precautions or warnings when using the code. Also, it will be specified if the code is version specific. The document mainly demonstrates using SQL and PL/SQL code to achieve any task. It does not concentrate on using OEM to perform a task. However, if there will be a significant advantage, there could be just some hints on using OEM for some tasks.

Prerequisites

The document assumes that the reader has already the knowledge of Oracle database administration.

How to Use the Document

1. Go to Contents section
2. Search the required task
3. Click on the required task link
4. Read the warnings and/or usage guideline, if any.
5. Make any modification in the code to match your case.

Oracle Database Versions

The code presented in the document is to operate on Oracle database versions 10g and 11g. It will be stated, if the code is version specific.

Obtaining Latest Version of the Document

Latest version can be obtained from my site or by emailing me at info@ahmedbaraka.com

Usage Terms

- Anyone is authorized to copy this document to any means of storage and present it in any format to any individual or organization for *non-commercial* purpose free.
- No individual or organization may use this document for *commercial* purpose without a written permission from the editor.
- This document is for informational purposes only, and may contain typographical errors and technical inaccuracies.
- **There is no warranty of any type for the code or information presented in this document. The editor is not responsible for any loses or damage resulted from using the information or executing the code in this document.**
- If any one wishes to correct a statement or a typing error or add a new piece of information, please send the request to **info@ahmedbaraka.com**

Document Parts

Part 1 Oracle DBA Fundamentals _____ 36

Part 2 Oracle Database Net Services _____ 238

Part 3 Oracle Database Backup and Recovery _____ 247

Part 4 Oracle Database Security _____ 311

Part 5 Oracle Database Performance Tuning _____ 340

Part 6 Oracle Automatic Storage Management (ASM) _____ 396

Part 7 Oracle Real Application Cluster _____ 410

Part 8 Oracle RAC One Node _____ 529

Part 9 Oracle Warehousing _____ 551

Part 10 Oracle Database Utilities _____ 556

Part 11 Miscellaneous Oracle Database Topics _____ 582

Part 12 PL/SQL Samples _____ 596

Part 13 Appendixes _____ 674

Contents

Part 1 Oracle DBA Fundamentals ____ **36**

DBA: Best Practices ____ **37**

DBA: Best Practice Guidelines for Standalone and RAC Databases ____ 37

Oracle Database Installation Consideration ____ **38**

Estimating Disk and Memoery Requirements ____ 38

Optimal Flexible Architecture ____ 38

Oracle Products Installed with the 11.1 Release ____ 39

Installing Oracle 10g R2 on Enterprise Linux 4 ____ **40**

Installation Environment ____ 40

Required Software ____ 40

Used Hardware ____ 40

Installation Plan ____ 40

1. Preinstallation tasks ____ 40
2. Oracle Database 10g Software Installation ____ 44
3. Apply Patchset 3 (10.2.0.4) for Clusterware and Database Software ____ 44
4. Configure Listeners ____ 45
5. Create Database ____ 45
6. Postinstallation Tasks ____ 45

Installing Oracle 11g R2 on Enterprise Linux 5 ____ **47**

Installation Environment ____ 47

Required Software ____ 47

Used Hardware ____ 47

Installation Plan ____ 47

1. Preinstallation tasks ____ 47
2. Oracle Database 11g Software Installation ____ 53
3. Apply Patchset ____ 53
4. Configure Listeners ____ 53
5. Create Database ____ 53
6. Postinstallation Tasks ____ 53

Installing Oracle 11g R2 on Enterprise Linux 5.5 with ASM _____ 54

Installation Environment _____54

Required Software _____54

Used Hardware _____54

Installation Plan _____54

1. Preinstallation tasks _____ 55
2. Oracle Grid Infrastructure installation_____ 60
3. Oracle Grid Infrastructure Patching_____ 61
4. Oracle Database 11g R2 Software Installation _____ 61
5. Oracle Database 11g R2 Software Patching _____ 62
6. Install EM Agent in cluster nodes (if required) _____ 62
7. ASM Diskgroups Creation _____ 62
8. Database Creation_____ 62
9. Postinstallation tasks_____ 62
10. General Useful Postinstallation Tasks in Linux _____ 62

Managing Oracle Database Instance _____ 64

Product Release Number _____64

Oracle Database Release Number Format_____ 64

Obtaining License Information_____64

Managing the Instance Architecture _____64

Obtaining Information about the Instance Processes _____ 64

Obtaining Information about the SGA_____ 65

Clearing the Buffer Cache _____ 65

Database Administration Authentication _____65

Using Operating System Authentication _____ 65

Using Password File Authentication_____ 66

Identifying Users SYSDBA or SYSOPER Users_____ 66

Data Dictionary and Dynamic Performance Views_____66

Data Dictionary Creation _____ 66

Startup and Shutdown_____66

Startup Levels _____ 66

Shutdown Levels _____ 66

Autostart of Database in Windows _____ 66

Automatically Starting Databases in Unix _____ 66

Quiescing a Database ____ 71
Suspending a Database ____ 71
Dropping a Database ____ 71
Initialization Files ____ 71
Managing Initialization Files ____ 71
Managing Parameters in SPFILE ____ 72
Alert and Trace Files ____ 72
Monitoring Alert and Trace Files ____ 72

Managing Oracle Database Physical Structure ____ 74

Managing Control Files ____ 74
Obtaining Control File information ____ 74
Creating Additional Copies, Renaming, and Relocating Control Files ____ 74
Creating New Control Files ____ 74
Backing Up Control Files ____ 75
Manage the Size of Control Files ____ 75
Multiplexing the Control File ____ 76
Maintaining Online Redo Log Files ____ 76
Forcing Log Switches and Checkpoints ____ 76
Adding Online Redo Log File Groups ____ 76
Adding Online Redo Log File Members ____ 76
Dropping Online Redo Log File Groups ____ 76
Dropping Online Redo Log File Members ____ 76
Relocating and Renaming Redo Log Members ____ 76
Verifying Blocks in Redo Log Files ____ 77
Clearing a Redo Log File ____ 77
Viewing Redo Log Information ____ 77
Managing Archived Redo Logs ____ 77
Obtaining Information about Archive Log ____ 77
Changing the Database Archiving Mode ____ 78
Specifying Archive Destinations and their Options ____ 78
Specifying the Minimum Number of Successful Destinations ____ 79
Controlling Archiving to a Destination ____ 79
Controlling Trace Output Generated by the Archivelog Process ____ 79
Managing Tablespaces ____ 79
Obtaining Tablespace Information ____ 79

Creating a Locally Managed Tablespace ____ 83
Specifying Segment Space Management____ 83
Adding Space to Tablespace ____ 83
Specifying Nonstandard Block Sizes for Tablespaces ____ 83
Using Bigfile Tablespace (BFT) ____ 83
Using Temporary Tablespace____ 84
Renaming a Tempfile____ 84
Shrinking Temporary Tablespace ____ 84
Using Default Temporary Tablespace ____ 85
Using Temporary Tablespace Groups ____ 85
Suppressing Redo Generation for a Tablespace ____ 85
Controlling Tablespaces Availability____ 85
Using Read-Only Tablespaces ____ 85
Renaming Tablespaces ____ 85
Default Permanent Tabelspace ____ 85
Dropping Tablespaces ____ 86
Managing the SYSAUX Tablespace ____ 86
Diagnosing and Repairing Locally Managed Tablespace Problems ____ 86
Verifying the Integrity of Segments Created in ASSM Tablespaces.____ 86
Checking Consistency of Segment Extent Map with Tablespace File Bitmaps____ 87
Verifying the Integrity of ASSM Tablespaces____ 88
Marking the Segment Corrupt or Valid ____ 89
Dropping a Corrupted Segment____ 89
Dumping a Segment Header and Bitmap Blocks____ 90
Marking a DBA Range in Bitmap as Free or Used ____ 90
Rebuilding the Appropriate Bitmap ____ 91
Rebuilding Quotas for Given Tablespace ____ 91
Migrating from a Dictionary-Managed to a Locally Managed Tablespace ____ 91
Fixing the State of the Segments in A Tablespace ____ 91
Scenario 1: Fixing Bitmap When Allocated Blocks are Marked Free (No Overlap) ____ 91
Scenario 2: Dropping a Corrupted Segment ____ 92
Scenario 3: Fixing Bitmap Where Overlap is Reported ____ 92
Scenario 4: Correcting Media Corruption of Bitmap Blocks____ 92
Transporting Tablespaces Between Databases ____ 92
Managing Alert Thresholds ____94
Getting the Current Threshold Setting ____ 94
Setting Tablespace Alert Thresholds ____ 95
Restoring a Tablespace to Database Default Thresholds____ 96

Modifying Database Default Thresholds ____ 97
Viewing Alerts ____ 97
Managing Datafiles and Tempfiles ____ 98
Creating Datafiles ____ 98
Enabling and Disabling Automatic Extension for a Datafile ____ 98
Manually Resizing a Datafile ____ 98
Bringing Datafiles Online or Taking Offline in ARCHIVELOG Mode ____ 98
Taking Datafiles Offline in NOARCHIVELOG Mode ____ 98
Renaming and Relocating Datafiles in a Single Tablespace ____ 99
Dropping Datafiles ____ 99
Copying a File on a Local File System ____ 99
Transferring a File to a Different Database ____ 100
Dumping a Data Block ____ 100
Managing Undo Tablespaces ____ 101
Obtaining Information on Undo ____ 101
Enabling Automatic Undo Management ____ 102
Creating an Undo Tablespace ____ 102
Setting Startup Undo Tablespace ____ 102
Tuning Undo Retention ____ 102
Using Undo Advisor ____ 103
Setting the Undo Retention Period ____ 103
Enabling Retention Guarantee ____ 103
Dropping an Undo Tablespace ____ 103
To Drop a Corrupt UNDO Tablespace ____ 104
Using Oracle Managed Files (OMF) ____ 104
Managing Schema Objects ____ 106
Chaching Small Tables in Memory ____ 106
Creating Virtual Columns ____ 106
Creating Partitioned Tables ____ 106
Partition Maintenance Operations ____ 111
Setting Deferred Segment Creation ____ 112
Creating Multiple Tables and Views in a Single Operation ____ 112
Collecting Object Statistics ____ 113
Collecting Index Statistics ____ 113
Collecting Table Statistics ____ 113

Collecting Schema Statistics ____ 114
Validating Tables, Indexes, Clusters, and Materialized Views ____115
Listing Chained and Migrated Rows of Tables and Clusters____115
Truncating Tables or Clusters ____116
Enabling and Disabling Triggers ____116
Managing Integrity Constraints ____116
Setting Constraint States and Deferability ____ 116
Modifying, Renaming, or Dropping Existing Integrity Constraints ____ 117
Reporting Constraint Exceptions ____ 117
Obtaining Information on Constraints ____ 118
Renaming Schema Objects ____118
Managing Object Dependencies____118
Manually Recompiling Views Procedures and Packages ____ 118
Switching to a Different Schema ____118
Using DBMS_METADATA to Display Information About Schema Objects ____118
Specifying Storage Parameters at Object Creation ____118
Managing Resumable Space Allocation ____119
Enabling Resumable Space Allocation ____ 119
Detecting Suspended Statements ____ 119
Obtaining Information about Suspended Statements ____ 121
Reclaiming Wasted Space ____121
Displaying Information About Space Usage for Schema Objects ____ 121
Segment Advisor ____ 124
Shrinking Database Segments Online ____ 126
Deallocating Unused Space ____ 126
Capacity Planning for Database Objects ____127
Estimating the Space Use of a Table ____ 127
Obtaining Object Growth Trends ____ 128
Using the SQL Access Advisor ____129
Estimating the Space Use of a Table ____ 129

Managing Tables ____ 134

Obtaining Information about Tables ____134
Creating Tables with some Options ____134

DML Error Logging _____135
Enabling Direct-Path INSERT _____135
Automatically Collecting Statistics on Tables _____136
Altering Tables _____136
Performing Online Redefinition with DBMS_REDEFINITION _____137
Redefining a Table _____ 137
Redefining a Single Partition _____ 139
Migrating BasicFile LOBs to SecureFiles _____ 140
Using Flashback Drop and Managing the Recycle Bin _____141
Managing Index-Organized Tables _____141
Managing External Tables _____142

Managing Indexes _____ 145

Using Indexes _____145
Using Bitmap Join Indexes (BJI) _____146
Partitioned Indexes _____147

Managing Materialized Views _____ 148

Obtaining Information about Materialized Views _____ 148
Monitoring the Progress of a Materialized View Refresh _____ 148
Materialized View Typical Refresh Errors _____ 152
Using Materialized Views _____ 152
Using Query Rewriting _____ 155
ReWrite Hints _____ 157
Using EXPLAIN_MVIEW Procedure: Viewing Materialized View Capabilities _____ 157
Using DBMS_ADVISOR.TUNE_MVIEW _____ 157
Registering a User-defined Table as Materialized View _____ 158

Managing Clusters and Hash Clusters _____ 159

Managing Views, Sequences, and Synonyms _____ 161

Managing Transactions _____ 162

Implementing Oracle's Concurrency Control _____162
Oracle Isolaction Levels _____ 162
Oracle Lock Types _____ 162

Identifying Blocking Sessions _____ 163
Using Autonomous Transaction _____163
Managing Long Transactions with Workspace Manager_____164

Repairing Corrupted Data _____ 172

Options for Repairing Data Block Corruption _____172
Detecting Corruptions Methods _____172
Using dbv (DBVerify) Utility_____172
Setting the Initialization Parameters for Detecting Corruption _____172
Verifying Block Integrity in Real Time: DB_BLOCK_CHECKING _____ 172
Verifying Block Integrity in Real Time: DB_BLOCK_CHECKSUM _____ 173
Detecting lost write: DB_LOST_WRITE_PROTECT_____ 173
Settubg the DB_ULTRA_SAFE Parameter (In Oracle 11g) _____ 173
Using ANALYZE Command_____173
Using EXP to Detect Corruption _____173
Using DBMS_REPAIR _____174
DBMS_REPAIR Limitations and Restrictions _____ 174
Evaluate the Costs and Benefits of Using DBMS_REPAIR _____ 174
Detect and Report Corruptions using DBMS_REPAIR _____ 175

Managing Automated Database Maintenance Tasks_____ 177

Predefined Automated Maintenance Tasks _____177
Predefined Maintenance Windows _____177
Obtaining Information about Predefined Maintenance Tasks_____177
Enabling and Disabling Maintenance Tasks _____178
Configuring Maintenance Windows _____178

Managing Resources_____ 180

Obtaining Information on Database Resource Manager _____191
Monitoring Oracle Database Resource Manager _____193

Using Oracle Scheduler_____ 195

Using Jobs _____195
Using Programs _____202

Using Schedules 203
Using Job Classes 204
Using Windows 205
Using Window Groups 206
Monitoring Job State with Email Notifications 207
Using File Watchers 209
Using Events Raised by the Scheduler 212
Using Events Raised by an Application (Events-Based Jobs) 214
Using Chains 217
Allocating Resources Among Jobs 221
Administering Oracle Scheduler 221
Configuring Oracle Scheduler 221
Monitoring and Managing the Scheduler 222
Enabling, Using and Disabling Remote External Jobs 223
Import/Export and the Scheduler 225
Scheduler Privileges 225
Scheduler Data Dictionary Views 225
Using the UTL_FILE Package 226

Data Loading and Transforming Tools 228

Using Database Links 229

Managing Diagnostic Data 230

Setting the Automatic Diagnostic Repository Directory 230
Using adrci Tool 230
General usage of adrci 230
Using adrci to Package Incidents 231
Managing Database Health Monitor 231
Managing Data Recovery Advisor 233
Using SQL Test Case Builder 233

Patching Oracle Products 235

Using Oracle Opatch 235

Part 2 Oracle Database Net Services _____ 238

Connectivity Naming Methods _____ 239

The Local Naming Method _____239

The Easy Connect Naming Method _____239

The External Naming Method_____239

The Directory Naming Method _____240

Database Resident Connection Pooling (DRCP) _____240

Oracle and Java Database Connectivity _____ 242

Establishing Database Connectivity _____242

Miscellaneous Connectivity Options _____ 244

Setting the Default Connect String _____244

Installing the Instant Client _____244

Setting Listener Options _____244

Setting Access Controls _____244

Changing Windows Hostname _____245

Part 3 Oracle Database Backup and Recovery _____ 247

Backup Guidelines _____ 248

Causes of Unplanned Down Time _____248

Causes of Planned Down Time_____248

Oracle's Solution to Down Time_____249

Minimizing Unplanned Downtime Guidelines _____249

SLA Sample _____250

Planning a Backup Strategy Guidelines _____250

Examples of Backup Schedules for a Database_____ 251

User-Managed Backups _____ 252

Obtaining Database File Information _____252

Making Whole Closed Database Backups _____ 252

Making a Whole Open Backup _____ 252

Making Tablespace Backups _____ 253
Obtaining Backup Status Information _____253
Checking Datafiles Taken as Backup _____253
Handling Crash Before User-Manged Backup Ends _____253
Backing up Control File _____254
Backing Up Initialization Files _____254

User-Managed Complete Recovery _____ 255

User-Managed Recovery in NOARCHIVELOG Mode _____255
User-Managed Recovery in NOARCHIVELOG Mode Without Redo Log File _____255
User-Managed Complete Recovery in ARCHIVELOG Mode _____255
Re-Creating Lost Datafiles Without Backup _____256

User-Managed Incomplete Recovery _____ 257

Common Situations Requiring Incomplete Recovery _____257
User-Managed Incomplete Recovery Steps _____257
Recovering from Lost Control File by Re-Creating the Control File _____257

Flash Recovery Area _____ 261

Obtaining Information on Flash Recovery Area _____261
Configuring Flash Recovery Area _____261
Backing Up the Flash Recovery Area _____261
Moving the Flash Recovery Area _____262

Recovery Manager (RMAN) _____ 263

Using A Media Management Layer (MML) with RMAN _____263
Obtaining Information about and related to RMAN using Dictionary Views _____263
Starting RMAN _____265
Using rlwrap Utility with RMAN in Unix-Based Systems _____266
Configuring the RMAN Environment _____266
RMAN Channel Commands _____267
Duration in days of RMAN information in Control File _____268
Monitoring RMAN Jobs _____268

Using RMAN BACKUP Command _____268
Backing Up Control File and SPFile _____269
Backing Up Archived RedLogs _____269
Backup in NOARCHIVELOG Mode_____270
Encrypting RMAN Backups_____270
Using Compression in RMAN Backups _____270
Using Multiplexed Backup Sets _____270
Using Parallelization of Backup Sets _____270
Using Duplexed Backup Sets (Backupset Copies) _____271
Making Image Copies_____271
Validating Backup _____272
Incremental Backup_____272
Tags for Backups and Image Copies _____272
Creating Archival Backups _____272
Monitoring RMAN Backups _____273
RMAN Complete Recovery _____273
Validating Backup Files _____ 273
Previewing Backup Files Required by a Restore _____ 274
Identifying Datafiles Requiring Recovery_____ 274
Performing Complete Recovery _____ 274
Restoring whole Database from RMAN Backups On a Different Node_____ 275
Restoring whole Database from RMAN Backups from A 32 bit to 64 bit_____ 278
RMAN Incomplete Recovery_____278
Simplified Recovery Through Resetlogs_____279
Recovering from Lost Control File using RMAN _____279
Block Media Recovery (BMR) _____280
Trial Recovery _____280
Handling Specific Errors During Recovery _____280
Configuring Instance Crash Recovery Time (MTTR) _____282
Working with the Data Recovery Advisor in RMAN _____282
RMAN Maintenance _____282
Cross Checking Backups and Copies _____ 282

Deleting Backups and Copies ____ 283
Changing the Availability of RMAN Backups and Copies ____ 283
Exempting a Backup or Copy from the Retention Policy ____ 283
The CATALOG Command ____ 284
The CHANGE … UNCATALOG Command ____ 284
RMAN Catalog ____ 284
Creating a Recovery Catalog ____ 284
RMAN Catalog Reporting ____ 285
Upgrading a Recovery Catalog ____ 285
Importing Recovery Catalogs ____ 285
Moving a Recovery Catalog ____ 286
Dropping a Recovery Catalog ____ 286
Virtual Private Catalogs ____ 286
Using RMAN Scripts ____ 287

Duplicating (Cloning) a Database ____ 289
Database Duplication (Cloning) Methods ____ 289
Database Duplication Techniques ____ 289
Database Duplication Prerequisites ____ 289
Duplicating an Active Database using RMAN ____ 289
Duplicating a Database without Recovery Catalog or Target Connection ____ 291
Manually Duplicating a Database ____ 293

Using Oracle Flashback Technology ____ 297

Flashback Options ____ 298

Preparing Your Database for Flashback ____ 299

Using Row Level Flashback Options ____ 300
Flashback Query ____ 300
Flashback Versions Query ____ 300
Flashback Transaction Query ____ 301
Flashback Transaction (Backout) ____ 301

Using Table Level Flashback Options ____ 303

Flashback Table _____303
Flashback Drop _____303
Flashback Data Archive _____304

Using Flashback Database _____ 308

When to use Flashback Database _____308
Flashback Database Considerations_____308
Using Flashback Database _____308
Restore Points _____309

Part 4 Oracle Database Security _____ 311

Oracle Database Security Management _____ 312

Security Guidelines_____ 313

Managing Users _____ 315

Database Authentication _____ 317

Managin Passwords _____317
External (OS) Authentication_____317
Proxy Authentication _____318
Logging In As a Different User_____318
Killing User Sessions from OS _____318

Controlling Database Access_____ 320

System and Object Privileges _____320
Invoker Rights and Definer Rights_____320
Roles _____320
Users, Roles, and Privileges Views_____321
Fine-Grained Data Access (Virtual Private Database VPD) _____322

Auditing Database _____ 325

Standard Auditing _____325
Customizing Database Auditing with Triggers _____326

Auditing the Database Using System Trigger _____328
Using Fine Grained Auditing_____331

Using Data Encryption _____ 334

Oracle Transparent Data Encryption (TDE) _____334
Tablespace Encryption _____335

Fine-Grained Access Control for UTL_* Packages _____ 337

Creating ACL _____337
Access Control Lists Maintenance _____338
Query Your Access Control List _____339

Part 5 Oracle Database Performance Tuning _____ 340

Managing Performance Statistics _____ 341

Managing OS Statistics _____341
Managing Database Statistics _____345
System and Session Statistics_____ 345
Time Model Statistics_____ 346
Wait Events _____ 347
Active Session History (ASH)_____ 349
Segment Statistics_____ 351
Handling Important Oracle Wait Events _____351
List of Idle Waits Events _____353
Using Performance Monitor Tool in Windows_____355
Using OS Watcher for Windows _____355

Optimizing Performance in Windows Server _____ 357

Optimizing Performance in Windows Server 2003_____357

Tuning the Database Instance _____ 358

Tuning the Shared Pool _____358
Tuning the Buffer Cache _____359
Tuning PGA _____361
Dumping the PGA _____361

Using Server Result Cache 362
Obtaining Information about Object Locks 364
Handling a Hanging Database 365
Accurately Measuring Process Size 365

Managing Automatic Workload Repository (AWR) 367

Managing Automated Maintenance Tasks 369
Using Automatic Database Diagnostic Monitor (ADDM) 369
Using Automatic SQL Tuning Advisor 371

Implementing Automatic Memory Management 374

Configuring DB_nK_CACHE_SIZE 376

Managing Optimizer Operations 377
Setting the Optimizer Mode 377
Defining Access Paths and Joins for the Query Optimizer 377
Gathering Optimizer Statistics 381
Gathering Object Statistics 381
Gathering System Statistics 382
Changing Statistics Preferences 382
Managing Pending and Published Statistics 383
Managing Extended Statistics 384
MultiColumn Statistics 384
Expression Statistics 385

A Simple Approach to Tuning SQL Statements 386

Using Application Tracing Tools 387
Using the SQL Trace Facility and TKPROF 387
Using the Event 10046 to Trace SQL Code 387
Tracing End to End Application 388
Enabling and Disabling Statistic Gathering for End to End Tracing 389

Writing Efficient SQL _____ **391**

Improving SQL Processing Techniques _____ **394**

Using SQL Tuning Advisor _____ **395**

Part 6 Oracle Automatic Storage Management (ASM) _____ **396**

Managing Oracle ASM _____ **397**

Obtaining Information about ASM Instance _____ 397

Creating an ASM Instance _____ 397

Managing Disk Groups and Disks in ASM _____ 398

Fundementals of Managing Disk Groups and Disks _____ 398

Managing Disk Groups Attributes _____ 399

Monitoring Long-Running Operations _____ 401

Migrating a Database to ASM _____ 402

Moving a Tablespace to ASM _____ 402

Accessing an ASM instance from DB Console _____ 403

Managing ASM Files _____ 404

Using ASMCMD Utility _____ 405

Using SYSASM Privilege and OSASM Group _____ 407

Manually Upgrading Oracle AS from 10g to 11g _____ 408

Verifying Manually ASM Device _____ 408

Part 7 Oracle Real Application Cluster _____ **410**

Oracle RAC Possible Installation Configurations _____ **411**

Installing Oracle 10g R2 RAC on Enterprise Linux 4 _____ **412**

Installation Environment _____ 412

Required Software _____ 412

Used Hardware _____ 412

Installation Plan _____ 412

1. Preinstallation tasks _____ 413

2. Oracle Clusterware installation _____ 420

3. Oracle Database 10g Software Installation ____ 424
4. Apply Patchset 3 (10.2.0.4) for Clusterware and Database Software ____ 425
5. Install EM Agent in cluster nodes (if required) ____ 426
6. Configure Listeners ____ 426
7. Perform ASM installation ____ 427
8. Perform cluster database creation ____ 427
9. Postinstallation tasks ____ 429
10. Useful Postinstallation Tasks ____ 430

Installing Oracle 11g R2 RAC on Enterprise Linux 5 ____ 431

Main Changes in Oracle 11g Release 2 RAC ____ 431
Installation Environment ____ 431
Required Software ____ 431
Used Hardware ____ 431
Installation Plan ____ 432
1. Preinstallation tasks ____ 433
2. Oracle Grid Infrastructure installation ____ 440
3. Oracle Grid Infrastructure Patching ____ 442
4. Oracle Database 11g R2 Software Installation ____ 442
5. Oracle Database 11g R2 Software Patching ____ 443
6. Install EM Agent in cluster nodes (if required) ____ 443
7. ASM Diskgroups Creation ____ 443
8. RAC Database Creation ____ 444
9. Postinstallation tasks ____ 445
10. General Useful Postinstallation Tasks in Linux ____ 446

Installing Oracle 10g R2 RAC on Windows ____ 447

Installation Methods ____ 447
Installation Environment ____ 447
Required Software ____ 447
Used Virtual Hardware ____ 447
Installation Plan ____ 448
1. Preinstallation tasks ____ 448
2. Oracle Clusterware installation ____ 450
3. Apply Patch Set 3 (10.2.0.4) on Clusterware software ____ 452
4. Oracle ASM 10g Software Installation ____ 453

5. Apply Patchset 3 (10.2.0.4) on ASM Software _____ 454
6. Install EM Agent in cluster nodes (if required) _____ 454
7. Configure Listeners _____ 454
8. Create ASM Instance _____ 455
9. Install Oracle RAC Database Home Software _____ 456
10. Apply Patchset 3 (10.2.0.4) on Oracle RAC Software Home _____ 457
11. Perform cluster database creation _____ 457
12. Useful Postinstallation Steps _____ 459

Cleaning Up Clusterware Installation on Windows _____ 460

Single Instance to RAC Conversion _____ 462

The Tools to Convert a Single Instance DB to RAC _____ 462
Conversion Prerequisites for Oracle 10g R2 _____ 462
Using rconfig Utitlity _____ 462
Using DBCA _____ 463

Administering RAC Database _____ 465

Administering Oracle Clusterware Components _____ 466

Managing Cluserware Daemons and Processes _____ 466
Displaying Clusterware Processes _____ 466
Starting, Stopping, Enabling and Disabling crs Stack _____ 466
CSS Parameters _____ 466
Administering Voting Disks in RAC _____ 466
Multiplexing Voting Disks _____ 466
Dynamically Adding and Removing Voting Disks after Installing RAC _____ 467
Backing up Voting Disks _____ 467
Recovering Voting Disks _____ 467
Administering the Oracle Cluster Registry (OCR) _____ 467
Replacing the OCR _____ 468
Adding and Removing the OCR _____ 468
Repairing the OCR _____ 468
Making Physical Backups of the OCR _____ 468
Recovering the OCR using the Physical Backups _____ 469
Making Logical Backups of the OCR (Exporting) _____ 469
Making Logical Backups of the OCR (Importing) _____ 469

Diagnosing OCR Problems with the OCRDUMP and OCRCHECK Utilities 470

Administering Storage 471

Datafile Access in Real Application Clusters 471

Redo Log File Storage in Real Application Clusters 471

Automatic Undo Management in Real Application Clusters 471

Administering ASM Instances with SRVCTL in RAC 472

Administering Cluster Databases 473

Displaying Current Instance in SQL*Plus Prompt 473

Starting and Stopping Instances and RAC Databases 473

Starting Up and Shutting Down with SQL*Plus 473

Intermittent Windows Shutdown Issue in RAC Environments 473

Starting Up and Shutting Down with SRVCTL 473

Customizing How Oracle Clusterware Manages RAC Databases 474

Switching Between the Database Automatic and Manual Policies 474

Customizing Resource Parameters (like AUTO_START) 474

Handling Initialization Parameter Files in RAC 475

Setting Server Parameter File Parameter Values for Real Application Clusters 475

Parameters Used in RAC Databases 475

Parameters that Must Have Identical Settings on All Instances 475

Parameters That Must Have Unique Settings on All Instances 476

Parameters that Should Have Identical Settings on All Instances 476

ASM Instance Initialization Parameters and RAC 476

Dropping a RAC Database 476

Workload Management in RAC 477

Types of Workload Distribution 477

Connection Load Balancing 477

Client-Side Load Balancing and Failover 477

Server-Side Load Balancing 477

Fast Application Notification (FAN) 478

Using Fast Application Notification Callouts 478

Configuring the Server-Side ONS 480

Administering Load Balancing Advisory 480

Monitoring Load Balancing Advisory _____ 481
Transparent Application Failover (TAF) _____ 482
TAF Basic Configuration without FAN (From Client Side) _____ 482
TAF Basic Configuration with FAN (Server-Side) _____ 482
TAF Preconnect Configuration _____ 482
Verifying TAF Configuration _____ 483
Enabling Distributed Transaction Processing for Services _____ 483

Administering Services _____ 484

Service Attributes _____ 484
Administering Services with DBCA _____ 484
Administering Services with PL/SQL _____ 484
Administering Services with SRVCTL _____ 486
Controlling the Preferred and Available Instances _____ 486
Using Services with Client Applications _____ 487
Services and the Scheduler _____ 487
Measuring Performance by Service Using the AWR _____ 488
Service Thresholds and Alerts _____ 490
Service Performance Views _____ 491
Restricted Session and Services _____ 491

Configuring Recovery Manager and Archiving _____ 492

Backup Possible Distributions in RAC _____ 492
RMAN Restore Scenarios for Real Application Clusters _____ 492
Cluster File System Restore Scheme _____ 492
Non-Cluster File System Restore Scheme _____ 492
RMAN and Oracle Net in Real Application Clusters _____ 492
Connecting to Specific Node _____ 492
Instance Recovery in Real Application Clusters _____ 493
Single Node Failure in Real Application Clusters _____ 493
Multiple-Node Failures in Real Application Clusters _____ 493
Configuring the RMAN Snapshot Control File Location _____ 493
Configuring the RMAN Control File and SPFILE Autobackup Feature _____ 493

Configuring Channels for RMAN in Real Application Clusters 493
Configuring Channels to use Automatic Workload Balancing 493
Configuring Channels to Use a Specific Instance 493
Node Affinity Awareness of Fast Connections 494
Archived Redo Log File Conventions in RAC 494
Archive Redo Log Configuration Scenarios 494
Automatic Storage Management and CFS Archiving Scheme 494
Non-Cluster File System Local Archiving Scheme 494
Changing the Archiving Mode in Real Application Clusters 495
Deleting Archived Redo Logs after a Successful Backup 495
Monitoring the Archiver Processes 495
Log_Archive_Dest_1 Set To Default Even When DB_Recovery_File_Dest Is Set (Bug 6373164) 495
Media Recovery in Real Application Clusters 497
Parallel Recovery in Real Application Clusters 497
Using a Flash Recovery Area in RAC 497

Managing Backup and Recovery 498

Administrative Options 499

Using Enterprise Manager Grid Control to Discover Nodes and Instances 499
Additional Information About SQL*Plus in RAC 499
How SQL*Plus Commands Affect Instances 499
Displaying Running Instances 499
Displaying Connect Identifier 499
Quiescing RAC Databases 500
Quiesced State and Cold Backups 500
Transparent Data Encryption and Wallets in RAC 500
Administering System and Network Interfaces with oifcfg 500
Defining Network Interfaces with oifcfg 500
Syntax and Commands for the oifcfg Command-Line Tool 500
Changing Public or Interconnect IP Subnet Configuration 501
Changing VIP Addresses 501

Adding Nodes and Instances on UNIX-Based Systems 503

Adding Nodes to a RAC Environment 503
Cloning Oracle Clusterware and RAC Software in Grid Environments 503
Quick-Start Node and Instance Addition Procedures 504
Adding an Oracle Clusterware Home to a New Node 504
Adding an Oracle Home with RAC to a New Node 505
Detailed Node and Instance Addition Procedure 505
Step 1: Connecting New Nodes to the Cluster 505
Step 2: Extending Clusterware and Oracle Software to New Nodes 505
Step 3: Preparing Storage on New Nodes 505
Step 4: Adding Nodes at the Oracle RAC Database Layer 506
Step 5: Adding Database Instances to New Nodes 506

Deleting Nodes and Instances on UNIX-Based Systems 507

Option 1: Quick-Start Node and Instance Deletion Procedures 507
Deleting an Oracle Home with RAC from an Existing Node 507
Deleting an Oracle Clusterware Home from an Existing Node 508
Option 2: Detailed Node and Instance Deletion Procedure 509
Step 1: Deleting DB Instances from Real Application Clusters Databases 509
Using Enterprise Manager to Delete Database Instances from Existing Nodes 509
Using DBCA in Interactive Mode to Delete Database Instances from Existing Nodes 509
Using DBCA in Silent Mode to Delete Instance from Existing Nodes 510
Step 2: Deleting Nodes from Real Application Clusters Databases 510
ASM Instance Clean-Up Procedures for Node Deletion 511

Adding and Deleting Nodes and Instances on Windows-Based Systems 512

Cloning Oracle Clusterware and RAC Software in Grid Environments 513
Quick-Start Node and Database Instance Addition and Deletion Procedures 513
Adding an Oracle Clusterware Home to a New Node 513
Adding an Oracle Home with RAC to a New Node 513
Deleting an Oracle Home with RAC from an Existing Node 513
Deleting an Oracle Clusterware Home from an Existing Node 513
Detailed Node and Database Instance Addition and Deletion Procedures 513
Overview of Node Addition Procedures 513

Step 1: Connecting New Nodes to the Cluster 513
Making Physical Connections 513
Installing the Operating System 513
Verifying the Installation with the Cluster Verification Utility 513
Checking the Installation 513
Step 2: Extending Oracle Software to New Nodes at the Oracle Clusterware 513
Step 3: Preparing Storage on New Nodes 513
Raw Device Storage Preparation for New Nodes 513
Step 4: Adding Nodes at the Oracle RAC Database Layer 513
Step 5: Adding Database Instances to New Nodes 513
Using Enterprise Manager to Add Database Instances to New Nodes 513
Using DBCA in Interactive Mode to Add Database Instances to New Nodes 513
Using DBCA in Silent Mode to Add Database Instances to New Nodes 513
Connecting to iSQL*Plus after Adding a Node 513
Adding Nodes that Already Have Clusterware and Oracle Software to a Cluster 514
Overview of Node Deletion Procedures 514
Step 1: Deleting Instances from Real Application Clusters Databases 514
Using Enterprise Manager to Delete Database Instances from Existing Nodes 514
Using DBCA in Interactive Mode to Delete Database Instances from Existing Nodes 514
Using DBCA in Silent Mode to Delete Instance from Existing Nodes 514
Step 2: Deleting Nodes from Real Application Clusters Databases 514
Step 3: ASM Instance Clean-Up Procedures for Node Deletion 514

Monitoring Performance 515

RAC Common Tuning Tips 515
Instance Recovery and RAC 515
Global Cache Wait Events 515
Monitoring Performance in Enterprise Manager 516
Using the Cluster Database Performance Page 516
Using the Cluster Database Instance Performance Page 516
Using the Cluster Performance Page 517
Using the Cluster Interconnects Page 517

Making Applications Highly Available Using Oracle Clusterware 518

Making an Application Highly Available Examples 518

Example1: Making an Application Highly Available ____ 518
Example2: Making an Application Highly Available ____ 520
Managing Automatic Oracle Clusterware Resource Operations for Action Scripts ____ 522
Displaying Clusterware Application and Application Resource Status Information ____ 522
Unregistering Applications and Application Resources ____ 523

RAC Troubleshooting ____ 524

Diagnosing the Oracle Clusterware High Availability Components ____524
Debugging Recommnedation____ 524
Clusterware Log Files and the Unified Log Directory Structure____ 524
Dynamic Debugging____ 524
Component Level Debugging____ 524
Oracle Clusterware Shutdown and Startup ____ 525
Enabling and Disabling Oracle Clusterware Daemons____ 525
Diagnostics Collection Script ____ 525
The Oracle Clusterware Alerts____ 525
Resource Debugging ____ 525
Checking the Health of the Clusterware ____ 525
Troubleshooting the Oracle Cluster Registry ____ 525
Troubleshooting Hostname Changes and CSS____ 525
Enabling Additional Tracing for Real Application Clusters High Availability____ 526
Diagnosing Oracle Real Application Clusters Components ____ 526
Where to Find Files for Analyzing Errors ____ 526
Using Instance-Specific Alert Files in Real Application Clusters ____ 526
Enabling Tracing for Java-Based Tools and Utilities in Real Application Clusters ____ 526
Resolving Pending Shutdown Issues ____ 526

Using the Cluster Verification Utility ____526
Cluster Verify Locations____ 526
Cluster Verify Stages____ 526
Cluster Verify Components ____ 527
CVU Component Verification Examples ____ 527
Understanding CVU Commands, Help, Output, and Nodelist Shortcuts ____ 528
Performing Various CVU Tests____ 528
Known Issues for the Cluster Verification Utility ____ 528

Part 8 Oracle RAC One Node ____ 529

Installing Oracle 11g R2 RAC One Node on Linux 5 ____ 530

Installation Environment _____ 530
Required Software _____ 530
Used Hardware _____ 530
Installation Plan _____ 531
1. Preinstallation tasks _____ 532
2. Oracle Grid Infrastructure installation _____ 539
3. Oracle Grid Infrastructure Patching _____ 542
4. Checking Oracle Grid Infrastructure Status _____ 542
5. Oracle Database 11g R2 Software Installation _____ 542
6. Oracle Database 11g R2 Software Patching _____ 543
7. Install EM Agent in cluster nodes (if required) _____ 544
8. ASM Diskgroups Creation _____ 544
9. RAC Database Creation _____ 545
10. Initialize the Database to RAC One Node (11.2.0.1 Only) _____ 547
11. Postinstallation tasks _____ 548
12. General Useful Postinstallation Tasks in Linux _____ 548

Instance Relocation using Omotion (11.2.0.1) _____ 549

Instance Relocation using Omotion (11.2.0.2) _____ 550

Part 9 Oracle Warehousing _____ 551

Oracle Warehouse Builder (OWB) _____ 552

Oracle Warehouse Builder Architecture _____ 552
Starting and Stoping the Service _____ 552
Configuring the Repository and Workspaces _____ 552
Steps of Using Warehouse Builder _____ 553
Mapping Operators _____ 554

Part 10 Oracle Database Utilities _____ 556

Using SQL*Plus _____ 557

Using SQL*Plus Command-Line Options _____ 557
Starting SQL*Plus Session _____ 557
Controlling User Privileges in SQL*Plus _____ 557

Setting the SQL*Plus Environment with the SET Command _____558
Setting SQL*Plus Preferances _____558
Logging SQL*Plus Errors _____559
Key SQL*Plus "Working" Commands_____559
Creating Command Files in SQL*Plus _____559
Copying Tables with the COPY Command _____559
Creating Web Pages Using SQL*Plus_____560
Using SQL to Generate SQL_____560
Enabling AUTOTRACE for a User _____560
Using rlwrap Utility with SQL*Plus in Unix-Based Systems_____561
Escaping Special Characters _____561

Using SQL*Loader Utility _____ 563

Invoking SQL*Loader_____563
Using SQL*Loader Control File _____563
Loading Excel File into a Table using SQL*Loader_____569
Loading Large Fields into a Table _____569
Using Direct Load Options _____570

Data Pump Export and Import _____ 571

Data Pump Components _____571
Data Pump Export Interfaces_____571
Export Modes Parameters _____571
Required Rrivileges _____571
Invoking Export Data Pump Examples_____571
Export Filtering Parameters_____572
Export Remapping Parameters _____572
Sampling Export Data _____573
Export Encryption Parameters_____573
Export Estimating Parameters_____573
Export Network Link Parameter_____573
Export PARALELL Parameter _____574

Import Modes Parameters _____574
File- and Directory-Related Parameters_____574
Using TABLE_EXISTS_ACTION Parameter _____574
Import Filtering Parameters_____574
Import Remapping Parameters _____574
Ignoring Nondeferred Constraints _____575
Import Network Link Parameter _____575
Import Flashback Parameters _____575
Import PARALELL Parameter _____576
Monitoring a Data Pump Jobs _____576

LogMiner_____ 577

Types of Supplemental Logging_____577
Levels of Supplemental Logging _____577
Disabling Database-Level Supplemental Logging _____577
LogMiner Dictionary Options _____578
Redo Log File Options _____578
OPTIONS possible values in DBMS_LOGMNR.START_LOGMNR: _____578
Obtaining LogMiner Operational Information_____578
Examples of Using LogMiner _____579
Without Sepecifying the Redo Files _____ 579
Wit Sepecifying the Redo Files _____ 580

Part 11 Miscellaneous Oracle Database Topics _____ 582

Managing Oracle Database Control_____ 583

Configuring and Using the Database Control_____583
Implementing EM Database Control Auto Startup_____583
Dropping and Recreating the Management Repository _____584

Installing Oracle 10g R5 (10.2) Enterprise Manager Grid Control for Linux x86 _____ 585

Installation Environment _____585

Required Software _____ 585
Used Hardware _____ 585
Installation Steps _____ 585

Remote Diagnostic Agent (RDA) _____ 593

Using Remote Diagnostic Agent (RDA) _____ 593

Connect Oracle to SQL Server _____ 594

Configuring and Using the Database Control _____ 594

Part 12 PL/SQL Samples _____ 596

PL/SQL Basics _____ 597

PL/SQL Data Types _____ 597
Controlling Compile-Time Displayed Warnings _____ 598
Catching Returned Errors _____ 599
Hiding Code _____ 599
Controlling Program Flow _____ 599

Using Cursors _____ 602

Using Records _____ 606

Using Table Functions _____ 608

Using Collections _____ 610

Using VARRAYS _____ 610
Using Nested Tables _____ 613
Using Associative Arrays _____ 614
Using Collection API _____ 616

Handling Errors _____ 621

Predefined Exceptions _____ 621
Using User-Defined Exceptions _____ 621
Using RAISE_APPLICATION_ERROR _____ 622

Autonomous Transactions ____ **624**

Some Stored Subprobrams Concepts ____ **625**

Serially Reusable Packages ____625

Stored Subprograms and Roles ____626

Invoker's vs. Definer's Rights ____626

Pinning an Programunit in the Shared Pool ____627

Using Triggers ____ **628**

Restrictions on Triggers ____ 628

Using DML Triggers ____628

Using Instead-of Triggers ____629

Using System Triggers ____630

Handling Mutating Tables in Triggers ____634

Dropping and Disabling Triggers ____635

Using Dynamic SQL ____ **636**

Working with Native Dynamic SQL ____636

Using DBMS_SQL ____639

Calling Java from PL/SQL ____ **648**

Configuring Oracle Database to Use External Routines ____ **649**

Using Large Objects (LOBs) ____ **652**

Creating LOB ____652

Using SQL with Internal LOBs ____652

Using LOBs in PL/SQL ____653

Performance Considerations ____662

Using Returning Clause ____ 662

Using CONTEXT Index ____ 663

Migrating from LONGs to LOBs ____664

PL/SQL Performance Tuning Tips ____ **665**

Use PL/SQL Profiler ____665

Use BULK COLLECT Clause _____666
Set PLSQL_OPTIMIZE_LEVEL and Subprogram Inlining _____670
Using Bind variables in Dynamic SQL _____671
Use NOCOPY Keyword _____672
Use Associative arrays _____672
Use Server Result Cache _____672

PL/SQL Miscellaneous Topics _____ 673

Accessing V$ Views from PL/SQL _____673

Part 13 Appendixes _____ 674

Program Units and Scripts Used in the Document _____ 675

Return Parameter Value for Normal User _____675
Applying Random Load on Database Sample 1 _____676
Setup _____ 676
Using the Load Generator Scripts _____ 682
Applying Random Load on Database Sample 2 _____683
Setup _____ 683
Using the Load Generator Scripts _____ 691

SQL Usage Samples _____ 693

Merge Command _____693
Multitable Inserts _____693
Parallel Insert _____694
Deleting Duplicate Rows in a Table _____694
Adding a Primary Key to a Table and then Filling it _____694

More DBA Scripts _____ 696

Compare Table Sizes in Two Databases _____696

Using Linux for Oracle _____ 697

Verifying the Kernel _____697
Checking for a Tainted Kernel _____697

Supported Hardware _____697
Using Oracle Relink Utility _____697
Certified and Supported File Systems _____698
Enterprise Linux Runlevels_____698
Using /etc/oratab File and dbstart Utility_____698
Automating Jobs _____699
Using cron _____ 699
Using anacron _____ 700
Using at command_____ 700
Using batch command _____ 701
Task Scheduler _____ 701
Configuring Linux Memory for Oracle _____ 701
Using Linux Performance Monitoring Tools _____701
About Linux Tools _____ 702
Using Linux Tools_____ 702
Checking Some General Guideline on Truning Oracle in Linux _____703
Troubleshooting Oracle Database in Linux _____704
Using OS Watcher (OSW) _____ 704
Using OS Watcher Graphs(OSWg)_____ 706
Using the On-Board Monitor (LTOM) _____ 706
Using strace_____ 707

Part 1 Oracle DBA Fundamentals

DBA: Best Practices

DBA: Best Practice Guidelines for Standalone and RAC Databases

- Use SPFILE
- Use temporary tablespaces
- Register all instances with remote listeners
- Use Database Resource Manager
- Use resumable space allocation
- Use Automatic Segment Space Management
- Use locally managed tablespaces
- Use Automatic Undo Management
- Enable block checking
- Enable Flashback Database
- Enable ARCHIVELOG mode and use a flash recovery area
- Use auto-tune checkpointing
- Log checkpoints to the alert log
- Multiplex production and standby redo logs
- Set CONTROL_FILE_RECORD_KEEP_TIME long enough
- Create two or more control files

Oracle Database Installation Consideration

Estimating Disk and Memoery Requirements

- Make sure the storage vendor is listed in the Oracle Storage Compatibility Program (OSCP).
- Allocate swap space that is about two to three times your Oracle RAM allocation.
- On Estimating the RAM:
  - OS RAM: 20 percent of total RAM for MS-Windows, 10% of RAM for UNIX
  - Each Oracle connection consumes two megabytes of RAM + sort_area_size + hash_area_size
  - If dynamic memory sizing is not used:
    - Shared_pool_size: If all the SQL statements that sent to ORACLE are using bind variable adequately, then 300M is enough in most cases
    - Large_pool_size: For dedicated Oracle server: 20-30M
    - Java_pool_size=10M
    - Data buffer: All the rest RAM should be allocated to Data buffer.

Optimal Flexible Architecture

- The OFA is a set of recommendations from Oracle Corporation aimed at simplifying management of complex software and databases often running under multiple versions of software.
- OFA Guidelines
  - Arrange at least four mount points that have names like /u01, /u02, /u03, and so on.
  - If the Oracle software owner is "oracle", make /u01/app/oracle your ORACLE_BASE

    ```
    # mkdir -p /u01/app/oracle
    # chown -R oracle:oinstall /u01/app/oracle
    # chmod -R 775 /u01/app/oracle
    ```

  - Make Oracle_HOME $ORACLE_BASE/product/version/{db|client|companion}_[n]. Examples:

    ```
    /u01/app/oracle/product/11.1.0/db_1
    /u01/app/oracle/product/11.1.0/client_1
    /u01/app/oracle/product/10.1.0.2.0/db_1
    ```

  - Install Flash Recovery Area in $ORACLE_BASE/flash_recovery_area in a mount point different from db files.
  - Administrative Files are to be created in ORACLE_BASE/admin/dbname
  - Store Oracle DB files in the mounted points.

    ```
    # mkdir /mount_point/oradata
    # chown oracle:oinstall /mount_point/oradata
    # chmod 775 /mount_point/oradata
    /u01/oradata/prod1/control01.ctl
    /u05/oradata/prod1/control02.ctl
    /u02/oradata/prod1/redo01.log
    /u04/oradata/prod1/redo02.log
    ```

    data file has this format:

    ```
    /u20/oradata/prod/system01.dbf
    ```

Oracle Products Installed with the 11.1 Release

- The following products are installed by default with database server 11g:
  - Oracle Application Express
  - Oracle Warehouse Builder
  - Oracle Configuration Manager: Tool that collects and uploads configuration information to the Oracle configuration repository
  - Oracle SQL Developer
  - Oracle Database Vault: Tool that enables you to secure business data

Installing Oracle 10g R2 on Enterprise Linux 4

Installation Environment

- Emulation software: VMWare Server 2 on for Windows
- OS: Oracle Linux Enterprise 4.5 for x86: kernel 2.6.9

Required Software

- Oracle Database 10g Release 2 for Linux x86 32-bit

Used Hardware

- In the VMWare: create one virtual machine (`oradb1`) with the following specs:
  - 2 GB RAM
  - an ethernet card
  - one local hardisk with 20 GB

Installation Plan

1. Preinstallation tasks:
   - Hardware requirements
   - Software requirements
   - Environment configuration
2. Oracle Database 10g Software Installation
3. Apply Patchset 3 (10.2.0.4) for Clusterware and Database Software
4. Configure Listeners
5. Create Database
6. Postinstallation tasks

1. Preinstallation tasks

- Install Oracle Enterprise Linux in the first local hardisk. Install nothing in the remaining disks.

  Note: for a production system, consider becoming an Oracle Unbreakable Linux customer and register your server on the Unbreakable Linux Network.

  - Give the ethernet card IP `192.168.4.21` and the hostname `oradb1.mydomain.com`. Define a gateway. If it does not exist, make it same as the host IP address.
  - Insall the following packages:
    - Desktop Environments
      - GNOME Desktop Environment

        - Desktop
            - X Window System
            - Gnome
        - Applications
            - Graphical Internet (optional)
        - Servers
            - Do not select anything in this group.
        - Development
            - Development Tools
        - System
            - Administration Tools
            - System Tools
                - Add the package 'sysstat' by clicking on the Details link and selecting "sysstat - The sar an iostat system monitoring commands." from the Optional Packages list.
        - Miscellaneous
            - Do not select anything in this group.
- Complete the installation
- Install further packages:

```
# to know distribution and version of Linux
cat /etc/issue
# to know kernel version (and its errata level)
uname -r

# from CD 3
rpm -Uvh libaio*
rpm -Uvh openmotif21-2.1.30-11.RHEL4.6.i386.rpm
rpm -Uvh openmotif-2.2.3-10.1.el4.i386.rpm

# those packages downloaded from http://rpm.pbone.net
rpm -e compat-libstdc++-296-2.96-132.7.2
rpm -Uvh compat-libstdc++-7.3-2.96.128.i386.rpm
rpm -Uvh compat-libstdc++-devel-7.3-2.96.128.i386.rpm
rpm -Uvh compat-gcc-7.3-2.96.128.i386.rpm
rpm -Uvh compat-gcc-c++-7.3-2.96.128.i386.rpm

# confirm the required packages are installed:
rpm -qa|grep gcc-
rpm -qa|grep glibc-
rpm -qa|grep compat-db-
rpm -qa|grep compat-gcc-
rpm -qa|grep compat-gcc-c++-
rpm -qa|grep compat-libstdc++-
rpm -qa|grep compat-libstdc++-devel-
rpm -qa|grep control-center-2.8.0
rpm -qa|grep openmotif21-
rpm -qa|grep setarch-

# SELINUX must be disabled
```

```
cat /etc/selinux/config | grep SELINUX=
vi /etc/selinux/config
SELINUX=disabled
shutdown -h now -r
```

- Check the hardware requirements

```
# Hardware Requirements
# At least 2 GB of physical memory
grep MemTotal /proc/meminfo

# swap space: twice the amount of physical memory
grep SwapTotal /proc/meminfo
# if you don't have enought swap,
#  you can add swap space by creating a temporary swap file.
#  let's say about 500MB:
dd if=/dev/zero of=tempswap bs=1k count=500000
chmod 600 tempswap
mke2fs tempswap
mkswap tempswap
swapon tempswap

# 400 MB disk space in /tmp
df -k /tmp
# 4 GB of disk space for Oracle software
df

The size of the shared memory should be at least the greater of
MEMORY_MAX_TARGET and MEMORY_TARGET for each Oracle instance on the computer.
To determine the amount of shared memory available, enter the following
command:
df -h /dev/shm/
# to adjust the shared memory file system size:
umount tmpfs
mount -t tmpfs shmfs -o size=1200m /dev/shm
```

- Create the required network configuration:

```
# Network names Resolution
# configure /etc/hosts if no domain server is used
cat /etc/hosts
127.0.0.1       localhost.localdomain   oradb1.mydomain.com     localhost
oradb1
```

- Create and configure the required OS users and groups

```
# inventory group
groupadd -g 501 oinstall
groupadd -g 502 dba
# oracle software owner user
/usr/sbin/useradd -u 200 -g oinstall -G dba oracle
passwd oracle
# make sure nobody user exists (if not there, create it useradd nobody)
id nobody

# The oracle User Environment
# in /home/oracle/.bash_profile
# export DISPLAY if required
export ORACLE_BASE=/u01/app/oracle
if [ $USER = "oracle" ]; then
```

```
 if [ $SHELL = "/bin/ksh" ]; then
  ulimit -p 16384
  ulimit -n 65536
 else
  ulimit -u 16384 -n 65536
 fi
 umask 022
fi
export EDITOR=vi
export ORACLE_HOME=$ORACLE_BASE/product/10.2.0/db_1
export ORACLE_PATH=$ORACLE_BASE/common/oracle/sql:.:$ORACLE_HOME/rdbms/admin
export ORACLE_SID=ora10g
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
export NLS_DATE_FORMAT="mm/dd/yyyy hh24:mi:ss"
export PATH=.:${PATH}:$HOME/bin:$ORACLE_HOME/bin:$ORA_CRS_HOME/bin
export PATH=${PATH}:/usr/bin:/bin:/usr/bin/X11:/usr/local/bin
export PATH=${PATH}:$ORACLE_BASE/common/oracle/bin
export ORACLE_TERM=xterm
export TNS_ADMIN=$ORACLE_HOME/network/admin
export ORA_NLS10=$ORACLE_HOME/nls/data
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$ORACLE_HOME/oracm/lib
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib:/usr/lib:/usr/local/lib
export CLASSPATH=$ORACLE_HOME/JRE
export CLASSPATH=${CLASSPATH}:$ORACLE_HOME/jlib
export CLASSPATH=${CLASSPATH}:$ORACLE_HOME/rdbms/jlib
export CLASSPATH=${CLASSPATH}:$ORACLE_HOME/network/jlib
export THREADS_FLAG=native
export TEMP=/tmp
export TMPDIR=/tmp

# User Shell Limits
# memlock is used to increase the per-process max locked memory
vi /etc/security/limits.conf
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft memlock 3145728
oracle hard memlock 3145728
```

- Configure kernel parameters and shell limits

Note: If you make a mistake with a parameter setting and your system does not start, then you must start Linux in the single-user runlevel (runlevel 1). At this runlevel, the /etc/sysctl.conf file is not run.

```
# Configuring kernel parameters and shell limits
# they can be tuned for a production db
# Append the following to the /etc/sysctl.conf file as the root user:
vi /etc/sysctl.conf
kernel.sem = 250 32000 100 128
# maximum size may be given to SGA (max 4GB)
# kernel.shmmax = 536870912 (512 M)
# following is 1 GB
kernel.shmmax =1073741824
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default = 4194304
net.core.rmem_max = 4194304
```

```
net.core.wmem_default = 262144
net.core.wmem_max = 262144
# to take immediate effect
/sbin/sysctl -p
```

- Configure hangcheck-timer kernel module:

```
# check hangcheck-timer Module Configuration
# with this module, if the kernel hangs, the machine will reboot
# verify the module is loaded
/sbin/lsmod | grep -i hang

# if not loaded, load it
vi /etc/modprobe.conf
options hangcheck-timer hangcheck_tick=30 hangcheck_margin=180
# execute and add in the file
vi /etc/rc.local
/sbin/modprobe hangcheck-timer
```

- Create the required directories for the Oracle database software:

```
# to know if there is an existing oracle inventory
# from its output, ORACLE_BASE will be parent of oraInventory
more /etc/oraInst.loc
# to identify existing Oracle home directories
more /etc/oratab

# in the example above, /u01 should be owned by the root user
#  and writable by group oinstall
cd /
chown -R oracle:oinstall /u01
chmod -R 775 /u01/oracle
mkdir -p /u01/app/oracle/product/10.2.0/db_1
mkdir /u01/stage
chown -R oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/oracle
chown oracle:oinstall /u01/stage
```

- As oracle, copy software installation into /u01/stage10g/db
- Disable screensavers on host & guest machines.
  - In Oracle Linux: Applications-> Preferences-> Screen Saver-> Mode: Disable Screen Saver
  - Do the same after logging off and logging on again as oracle user.
- Restart the machine.

2. Oracle Database 10g Software Installation

```
-- start OUI
su - oracle
cd /u01/stage10g/db
./runInstaller

Follow the steps.
Install database software only.
```

3. Apply Patchset 3 (10.2.0.4) for Clusterware and Database Software

```
# extract 10g Release 2 (10.2.0.4) Patch Set 3 for Linux x86 to
```

```
/u01/stage10g/patch10.2.0.4/Disk1
mkdir /u01/stage10g/patch10.2.0.4

cd /u01/stage10g/patch10.2.0.4/Disk1
./runInstaller
Select path of  ORACLE_HOME ->Next
```

4. Configure Listeners

```
cd /u01/app/oracle/product/10.2.0/db_1/bin
./netca &
Add a new listener

-- optionally, use net manager to manually register the database:
./netmgr
```

5. Create Database

```
cd /u01/app/oracle/product/10.2.0/db_1/bin
./dbca &
follow the steps to create a customized database of SID "ora10g" and Global
nmae "ora10g.oradb1".
Do not use specific templates (non-Custom).

The DBCA may pop up the following message in the end of db creation:
"Error securing Database control, Database Control has been brought up in non-
secure mode. To secure the Database Control execute the following commands:
... "

# check the Oracle processes:
ps -eo pid -o command | grep ora_ | grep -v grep
```

6. Postinstallation Tasks

- Verify OEM:

```
# verify that OEM is working
http://oradb1:1158/em
# restart the dbconsole if required
emctl status dbconsole
emctl stop dbconsole
emctl start dbconsole
```

- Verify iSQL*Plus:

```
# in the browser
http://oradb1:5560/isqlplus

# if not running:
isqlplusctl start
```

- Consider implementing automatic database startup. See: Automatically Starting Databases
- Consider implementing automatic EM Database Control startup. See: Implementing EM Database Control Auto Startup
- Consider using rlwrap utility with SQL*Plus and RMAN:
    - Using rlwrap Utility with RMAN in Unix-Based Systems
    - Using rlwrap Utility with SQL*Plus in Unix-Based Systems
- For easy Oracle Home access:

```
echo "alias db='cd $ORACLE_HOME'" >> /home/oracle/.bashrc
```

Installing Oracle 11g R2 on Enterprise Linux 5

Note: for any installation, you should check the Release Notes documenation before taking any practical step.

Installation Environment

- Emulation software: VMWare Workstation 7 for Windows
- OS: Red Hat Enterprise Linux 5.2 for x86

Required Software

- Oracle Database 11g Release 2 for Linux x86 32-bit

Used Hardware

- In the VMWare: create one virtual machine with the following specs:
  - 2.0 GB RAM
  - One ethernet cards: can be configured as bridged or host-only in VMware.
  - One local hardisk with 32 GB on SCSI 0:0. It will be used for software installation.
  - One local hardisk with 20 GB on SCSI 1:0. It will be used for Oracle Database data files.
  - One local hardisk with 20 GB on SCSI 1:1. It will be used for Oracle Database flash recovery.
  - CPU Count: 2 (optional)

Installation Plan

1. Preinstallation tasks:
   - Hardware requirements
   - Software requirements
   - Environment configuration
2. Oracle Database 11g Software Installation
3. Apply Patchset
4. Configure Listeners
5. Create Database
6. Postinstallation tasks

1. Preinstallation tasks

- Install Oracle Enterprise Linux in the first local hardisk. Install nothing in the remaining disks.

Note: for a production system, consider becoming an Oracle Unbreakable Linux customer and register your server on the Unbreakable Linux Network.

- Give the ethernet card IP `192.168.4.100` and the hostname `srv100.mydomain.com`. Define a gateway. If it does not exist, make it same as the host IP address.
- Insall the following packages:
    - Desktop Environments
        - GNOME Desktop Environment
    - Applications
        - Graphical Internet (optional)
        - Editors (optional)
    - Development
        - Development Libraries
        - Development Tools
    - Servers
        - Do not select anything in this group.
    - Base System
        - Administration Tools
        - System Tools
            - Add the package 'sysstat' by clicking on the Details link and selecting "sysstat - The sar an iostat system monitoring commands." from the Optional Packages list.
    - X Window System

- Complete the installation
- RHEL 5 Bug: After the Installation compelets, RHEL 5.2 and below will hang on booting when it reaches to "starting udev" line. To solve this problem, shutdown the Vmware machine and change the CPU count and Core Count to only one. Implement the changes below, then shutdown the machine, set CPU count back to 2 and startup the machine.

  put the kernel command line parameters at the end of the "kernel" line:

  `vi /boot/grub/grub.conf`

  add `divider=10 clocksource=acpi_pm`

  For example: `kernel /vmlinuz-2.6.18 .. clock=acpi_pm divider=10`

- For Vmware machines, install VMWare tools.
- Install further packages:

```
# to know distribution and version of Linux (Red Hat Ent. 5.2 used)
cat /etc/issue
# to know kernel version (and its errata level) (2.6.18-92 or newer)
uname -r

# to list missed packages:
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n' binutils \
compat-libstdc++-33 \
elfutils-libelf \
elfutils-libelf-devel \
gcc \
```

```
gcc-c++ \
glibc \
glibc-common \
glibc-devel \
glibc-headers \
ksh \
libaio \
libaio-devel \
libgcc \
libstdc++ \
libstdc++-devel \
make \
sysstat \
unixODBC \
unixODBC-devel

# for missed packages, install them:
rpm -Uvh libaio-devel-0.3.106-3.2.i386.rpm
rpm -Uvh unixODBC*


# SELINUX must be disabled
cat /etc/selinux/config | grep SELINUX=
vi /etc/selinux/config
SELINUX=disabled
shutdown -h now -r
```

- Check the hardware requirements

```
# Hardware Requirements (in cluster nodes)
# At least 1.0 GB of physical memory
grep MemTotal /proc/meminfo

# swap space: same as the amount of physical memory
grep SwapTotal /proc/meminfo

# to display swap and memory in one command:
free

# if you don't have enought swap,
#  you can add swap space by creating a temporary swap file.
#  let's say about 500MB:
dd if=/dev/zero of=tempswap bs=1k count=500000
chmod 600 tempswap
mke2fs tempswap
mkswap tempswap
swapon tempswap

# 1 GB disk space in /tmp
df -h /tmp
# 4 GB of disk space for Oracle software
df

The size of the shared memory should be at least the greater of
MEMORY_MAX_TARGET and MEMORY_TARGET for each Oracle instance on the computer.
To determine the amount of shared memory available, enter the following
command:
df -h /dev/shm/
```

```
# to adjust the shared memory file system size:
umount tmpfs
mount -t tmpfs shmfs -o size=1200m /dev/shm
```

- Create the required network configuration:

```
ping srv100
ping srv100.mydomain.com

# Network names Resolution
# configure /etc/hosts if no domain server is used
cat /etc/hosts
127.0.0.1       srv100.mydomain.com     srv100  localhost.localdomain
localhost
```

- Create and configure the required OS users and groups

```
# all group and user ids on all the nodes must have identical id
# Grid Infrastructure (GI) and the Oracle RDBMS home will
# be installed using different users:
# oracle inventory group
/usr/sbin/groupadd -g 501 oinstall
/usr/sbin/groupadd -g 502 dba
/usr/sbin/useradd -u 502 -g oinstall -G dba oracle

# set passwords
passwd oracle

# make sure nobody user exists (if not there, create it useradd nobody)
id nobody


# define the env variables for oracle user
vi /home/oracle/.bash_profile
# Oracle evn vars
export EDITOR=vi
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=srv100.mydomain.com
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0/db_1
export ORACLE_SID=ora11gr2
export ORACLE_TERM=xterm
export PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib


# shell startup file
vi /etc/profile
if [ $USER = "oracle" ] || [ $USER = "grid" ]; then
 if [ $SHELL = "/bin/ksh" ]; then
  ulimit -p 16384
  ulimit -n 65536
 else
  ulimit -u 16384 -n 65536
 fi
 umask 022
fi
```

```
# for C shell
vi /etc/csh.login
if ( $USER = "oracle" || $USER = "grid" ) then
 limit maxproc 16384
 limit descriptors 65536
endif
```

- Configure kernel parameters and shell limits

Note: If you make a mistake with a parameter setting and your system does not start, then you must start Linux in the single-user runlevel (runlevel 1). At this runlevel, the /etc/sysctl.conf file is not run.

```
# Configuring kernel parameters and shell limits
# they can be tuned for a production db
# Append the following to the /etc/sysctl.conf file as the root user:
vi /etc/sysctl.conf

# kernel.shmmax not stated in 11g R2 (max: 4GB) (169706.1)
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.aio-max-nr = 1048576
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576

# to take immediate effect
/sbin/sysctl -p

# User Shell Limits
# memlock is used to increase the per-process max locked memory
vi /etc/security/limits.conf
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536

vi /etc/pam.d/login
session required pam_limits.so
```

- Create the required directories for the Oracle database software:

```
# to know if there is an existing oracle inventory
# from its output, ORACLE_BASE will be parent of oraInventory
more /etc/oraInst.loc
# to identify existing Oracle home directories
more /etc/oratab

# Oracle Inventory Directory
# as a root
mkdir -p /u01/app/oraInventory
chown -R oracle:oinstall /u01/app/oraInventory
chmod -R 775 /u01/app/oraInventory

# Oracle Base Directory
mkdir -p /u01/app/oracle
```

```
#needed to ensure that dbca is able to run after the rdbms installation
mkdir -p /u01/app/oracle/cfgtoollogs
chown -R oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/app/oracle

# Oracle RDBMS Home Directory
mkdir -p /u01/app/oracle/product/11.2.0/db_1
chown -R oracle:oinstall /u01/app/oracle/product/11.2.0/db_1
chmod -R 775 /u01/app/oracle/product/11.2.0/db_1

# staging area to hold software installation
mkdir -p /u01/stage11g/db
chown -R oracle:oinstall /u01/stage11g/db
chmod -R 775 /u01/stage11g/db
```

- Partition and mount the disks:

```
#define the disks to partition
fdisk -l | grep '^Disk'

# as a root, for the disks /dev/sdb and /dev/sdc
#partition the disks:
fdisk /dev/sdb
# answers: "n", "p", "1", "Return", "Return", "p" and "w"
Note: if the following message appears after the "w" command:
 WARNING: Re-reading the partition table failed with error 16: Device or
 resource  busy, then you can avoid restarting the machine by the following
command: partprobe

# to make sure partions are created
ls -lX /dev/sd*

# format the paritions
mkfs.ext3 /dev/sdb1
mkfs.ext3 /dev/sdc1

# Mount the new disk
mkdir /u02
mount /dev/sdb1 /u02
mkdir /u03
mount /dev/sdc1 /u03
df -H

# Update /etc/fstab
vi /etc/fstab
/dev/sdb1               /u02           ext3    defaults        1 2
/dev/sdc1               /u03           ext3    defaults        1 2

# create folder for the db data
mkdir -p /u02/oradata/
chown -R oracle:oinstall /u02/oradata/
chmod -R 775 /u02/oradata/

# create folder for the flash area
mkdir -p /u03/oraflash/
chown -R oracle:oinstall /u03/oraflash/
chmod -R 775 /u03/oraflash/
```

- As **oracle**, copy software installation into /u01/stage11g/db

2. Oracle Database 11g Software Installation

```
-- start OUI
su - oracle
cd /u01/stage11g/db
./runInstaller

Follow the steps.
Install database software only.
```

3. Apply Patchset

```

```

4. Configure Listeners

```
cd /u01/app/oracle/product/10.2.0/db_1/bin
./netca &
Add a new listener
Add Naming mtehods: Local Naming, EZConnect
```

5. Create Database

```
cd /u01/app/oracle/product/10.2.0/db_1/bin
./dbca &
follow the steps to create a database with sid: ora11gr2

The DBCA may pop up the following message in the end of db creation:
"Error securing Database control, Database Control has been brought up in non-
secure mode. To secure the Database Control execute the following commands:
... "
```

6. Postinstallation Tasks

- Verify OEM:

```
# verify that OEM is working
https://srv100.mydomain.com:1158/em
# restart the dbconsole if required
emctl status dbconsole
emctl stop dbconsole
emctl start dbconsole

# check Oracle processes:
ps -eo pid -o command | grep ora_ | grep -v grep
```

- Consider implementing automatic database startup. See: Automatically Starting Databases
- Consider implementing automatic EM Database Control startup. See: Implementing EM Database Control Auto Startup
- Consider using rlwrap utility with SQL*Plus and RMAN:
    - Using rlwrap Utility with RMAN in Unix-Based Systems
    - Using rlwrap Utility with SQL*Plus in Unix-Based Systems
- For easy Oracle Home access:

```
echo "alias db='cd $ORACLE_HOME'" >> /home/oracle/.bashrc
```

Installing Oracle 11g R2 on Enterprise Linux 5.5 with ASM

Note: for any installation, you should check the Release Notes documenation before taking any practical step.

Installation Environment

- Emulation software: VMWare Workstation 7 for Windows
- OS: Oracle Enterprise Linux 5.5 for x86 64-bit

Required Software

- Oracle Database 11g Release 2 for Linux x86 64-bit
- Oracle Database 11g Release 2 Grid Infrastructure (11.2.0.1.0) for Linux x86 64-bit

Used Hardware

- In the VMWare: create one virtual machine with the following specs:
  - 2.5 GB RAM
  - One ethernet card configured as bridged or host-only in VMware
  - CPU Count: 2
  - Disk1: 34 GB on SCSI 0:0 used to install the OS and software
  - Disk2: of 12 GB. It will be used for +Data. Set it on controller SCSI 1:1
  - Disk3: of 6 GB. It will be used for +Flash. Set it on controller SCSI 1:2

Installation Plan

1. Preinstallation tasks
   - Hardware requirements
   - Software requirements
   - Environment configuration
2. Oracle Grid Infrastructure installation
3. Oracle Grid Infrastructure Patching
4. Oracle Database 11g R2 Software Installation
5. Oracle Database 11g R2 Software Patching
6. Install EM Agent in cluster nodes (if required)
7. ASM Diskgroups Creation
8. Database Creation
9. Complete postinstallation tasks

10. Useful postinstallation tasks

1. Preinstallation tasks

- Install Oracle Enterprise Linux in the first local hardisk. Install nothing in the remaining disks.

  Note: for a production system, consider becoming an Oracle Unbreakable Linux customer and register your server on the Unbreakable Linux Network.

  - Configure the swap area in the local hardisk to have 6 GB disk space.
  - Give the ethernet card IP 192.168.4.105 the hostname `srv07`
  - Insall the following packages:
    - Desktop Environments
      - GNOME Desktop Environment
    - Applications
      - Graphical Internet (optional)
      - Editors (optional)
    - Development
      - Development Libraries
      - Development Tools
    - Servers
      - Do not select anything in this group.
    - Base System
      - Administration Tools
      - System Tools
        - Add the package 'sysstat' by clicking on the Details link and selecting "sysstat - The sar an iostat system monitoring commands." from the Optional Packages list.
    - X Window System
- Complete the installation
- Install further packages:

```
# to know distribution and version of Linux
cat /etc/issue
# to know kernel version (and its errata level) (2.6.18-194.el5)
uname -r

# to list missed packages:
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n' binutils \
compat-libstdc++-33 \
elfutils-libelf \
elfutils-libelf-devel \
gcc \
gcc-c++ \
glibc \
glibc-common \
glibc-devel \
```

```
glibc-headers \
ksh \
libaio \
libaio-devel \
libgcc \
libstdc++ \
libstdc++-devel \
make \
sysstat \
unixODBC \
unixODBC-devel

# for missed packages, install them:
rpm -Uvh elfutils-libelf-0.137-3.el5.i386.rpm
rpm -ivh elfutils-libelf-devel-static-0.137-3.el5.i386.rpm elfutils-libelf-
devel-0.137-3.el5.i386.rpm
rpm -Uvh kernel-headers-2.6.18-194.el5.x86_64.rpm
rpm -Uvh glibc-headers-2.5-49.x86_64.rpm
rpm -Uvh glibc-devel-2.5-49.i386.rpm
rpm -Uvh glibc-devel-2.5-49.x86_64.rpm
rpm -Uvh libgomp-4.4.0-6.el5.x86_64.rpm
rpm -Uvh gcc-4.1.2-48.el5.x86_64.rpm
rpm -Uvh gcc-c++-4.1.2-48.el5.x86_64.rpm
rpm -Uvh libstdc++-devel-4.1.2-48.el5.x86_64.rpm
rpm -Uvh gcc-c++-4.1.2-48.el5.x86_64.rpm
rpm -Uvh libaio-devel-0.3.106-5.i386.rpm
rpm -Uvh libaio-devel-0.3.106-5.x86_64.rpm
rpm -Uvh unixODBC-2.2.11-7.1.i386.rpm
rpm -Uvh unixODBC-2.2.11-7.1.x86_64.rpm
rpm -Uvh unixODBC-devel-2.2.11-7.1.i386.rpm
rpm -Uvh unixODBC-devel-2.2.11-7.1.x86_64.rpm


# Oracle ASM Libaray and drivers can be downloaded from here
# to know the kernel verion: uname -rm
# In this case we need:
# library and tools
rpm -Uvh oracleasm-support-2.1.3-1.el5.x86_64.rpm
rpm -Uvh oracleasm-2.6.18-194.el5-2.0.5-1.el5.x86_64.rpm
rpm -Uvh oracleasmlib-2.0.4-1.el5.x86_64.rpm

# SELINUX must be disabled
cat /etc/selinux/config | grep SELINUX=
vi /etc/selinux/config
SELINUX=disabled
shutdown -h now -r
```

- Check the hardware requirements

```
# Hardware Requirements (in cluster nodes)
# At least 1.5 GB of physical memory but practically 1.5 is not fine
grep MemTotal /proc/meminfo

# swap space: same as the amount of physical memory
grep SwapTotal /proc/meminfo

# to display swap and memory in one command:
free
```

```
# if you don't have enought swap,
#  you can add swap space by creating a temporary swap file.
#  let's say about 500MB:
dd if=/dev/zero of=tempswap bs=1k count=500000
chmod 600 tempswap
mke2fs tempswap
mkswap tempswap
swapon tempswap

# 1 GB disk space in /tmp
df -h /tmp
# 8 GB of disk space for Oracle software
df

The size of the shared memory should be at least the greater of
MEMORY_MAX_TARGET and MEMORY_TARGET for each Oracle instance on the computer.
To determine the amount of shared memory available, enter the following
command:
df -h /dev/shm/
# to adjust the shared memory file system size:
umount tmpfs
mount -t tmpfs shmfs -o size=1200m /dev/shm
```

- Create the required network configuration:

```
ping srv07

# Network names Resolution
cat /etc/hosts
127.0.0.1               srv07 localhost.localdomain localhost
```

- Create and configure the required OS users and groups

  Note: When I tried using differenct OS users and groups for Grid Infrastrucutre and ASM, I got error later in DBCA.

```
# Grid Infrastructure (GI) and the Oracle RDBMS home will
# be installed using different users:
/usr/sbin/groupadd -g 501 oinstall
/usr/sbin/groupadd -g 502 dba
/usr/sbin/groupadd -g 504 asmadmin
/usr/sbin/useradd -u 502 -g oinstall -G dba,asmadmin oracle

# set passwords
passwd oracle

# make sure nobody user exists (if not there, create it useradd nobody)
id nobody

# define the env variables for oracle user
vi /home/oracle/.bash_profile

# Oracle Settings
export EDITOR=vi
TMP=/tmp; export TMP
TMPDIR=$TMP; export TMPDIR
ORACLE_HOSTNAME=srv07; export ORACLE_HOSTNAME
ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
```

```
ORACLE_HOME=$ORACLE_BASE/product/11.2.0/db_1; export ORACLE_HOME
ORACLE_SID=oradb; export ORACLE_SID
ORACLE_TERM=xterm; export ORACLE_TERM
PATH=/usr/sbin:$PATH; export PATH
PATH=$ORACLE_HOME/bin:$PATH; export PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export LD_LIBRARY_PATH
CLASSPATH=$ORACLE HOME/jdk/jre/lib/:$ORACLE HOME/jlib:$ORACLE HOME/rdbms/jlib;
export CLASSPATH

# shell startup file
vi /etc/profile
if [ $USER = "oracle" ] || [ $USER = "grid" ]; then
 if [ $SHELL = "/bin/ksh" ]; then
  ulimit -p 16384
  ulimit -n 65536
 else
  ulimit -u 16384 -n 65536
 fi
 umask 022
fi

# for C shell
vi /etc/csh.login
if ( $USER = "oracle" || $USER = "grid" ) then
 limit maxproc 16384
 limit descriptors 65536
endif
```

- Configure kernel parameters and shell limits

  Note: If you make a mistake with a parameter setting and your system does not start, then you must start Linux in the single-user runlevel (runlevel 1). At this runlevel, the /etc/sysctl.conf file is not run.

```
# Kernel Parameters
# to tune thme, refer to metalink document 169706.1
# Append the following to the /etc/sysctl.conf file as the root user:
vi /etc/sysctl.conf

kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.aio-max-nr = 1048576
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576

# to take immediate effect
/sbin/sysctl -p

# User Shell Limits
# memlock is used to increase the per-process max locked memory
vi /etc/security/limits.conf

oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
```

```
vi /etc/pam.d/login
session required pam_limits.so
```

- Create partitions in the disks:

Note: On a real life storage, you would create a single whole-disk partition with exactly 1 MB offset on each LUN to be used as ASM Disk. In fdisk: u (to change units from cylinder to sectors), n, p, 1, 2048, w.

```
# as a root, for the disks /dev/sdb .. /dev/sdd
# confirm they are seen:
ls /dev/sd*
#partition the disks:
fdisk /dev/sdb
# answers: "n", "p", "1", "Return", "Return", "p" and "w"
Note: if the following message appears after the "w" command:
 WARNING: Re-reading the partition table failed with error 16: Device or
 resource  busy, then you can avoid restarting the machine by the following
command: partprobe

# to make sure partions are created
ls -lX /dev/sd*
```

- Configure ASM drivers:

```
# as root  (to be done in all nodes)
oracleasm configure -i

Default user to own the driver interface []: oracle
Default group to own the driver interface []: asmadmin
Start Oracle ASM library driver on boot (y/n) [n]: y
Fix permissions of Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration: done

# Load the kernel module using the following command:
/usr/sbin/oracleasm init
# If you have any problems, make sure you have the correct
#  version of the driver (may require Internet connection):
/usr/sbin/oracleasm update-driver

# mark the shared disks: (one node)
/usr/sbin/oracleasm createdisk DISK1 /dev/sdb1
/usr/sbin/oracleasm createdisk DISK2 /dev/sdc1

# check the disks are marked and seen:
/usr/sbin/oracleasm listdisks

#If you need to unmark a disk that was used in a createdisk command:
/usr/sbin/oracleasm deletedisk DISK1
/usr/sbin/oracleasm deletedisk DISK2
```

- Create the required directories for the Oracle software:

```
# to know if there is an existing oracle inventory
# from its output, ORACLE_BASE will be parent of oraInventory
more /etc/oraInst.loc
# to identify existing Oracle home directories
more /etc/oratab
```

```
# Oracle Inventory Directory
# as a root
mkdir -p /u01/app/oraInventory
chown -R oracle:oinstall /u01/app/oraInventory
chmod -R 775 /u01/app/oraInventory

# Grid Infrastructure Home Directory
mkdir -p /u01/11.2.0/grid
chown -R oracle:oinstall /u01/11.2.0/grid
chmod -R 775 /u01/11.2.0/grid

# Oracle Base Directory
mkdir -p /u01/app/oracle
chown -R oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/app/oracle

#needed to ensure that dbca is able to run after the rdbms installation
mkdir /u01/app/oracle/cfgtoollogs
chown -R oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/app/oracle

# Oracle RDBMS Home Directory
mkdir -p /u01/app/oracle/product/11.2.0/db_1
chown -R oracle:oinstall /u01/app/oracle/product/11.2.0/db_1
chmod -R 775 /u01/app/oracle/product/11.2.0/db_1
```

2. Oracle Grid Infrastructure installation

```
# software staging folder
mkdir -p /u01/app/stage/ora11gr2gridinfra
chown -R oracle:oinstall /u01/app/stage/ora11gr2gridinfra
chmod -R 775 /u01/app/stage/ora11gr2gridinfra

mkdir -p /u01/app/stage/ora11gr2db
chown -R oracle:oinstall /u01/app/stage/ora11gr2db
chmod -R 775 /u01/app/stage/ora11gr2db

# as oracle: copy Grid software files to /u01/app/stage/ora11gr2db

# as root:
chmod -R 775 /u01/app/stage/ora11gr2gridinfra

# lunch OUI from the clusterware ( as oracle)
cd /u01/app/stage/ora11gr2gridinfra
./runInstaller

Installation Option
>Select radio button 'Install and Configure Grid Infrastructure for a
Standalone Server'
>Next

Product Language
>Accept 'English' as language'
>Next
Creat ASM Disk Group
>Disk Group Name: Data (12GB disk: DISK1)
>Redundancy: external
>Next
```

```
NOTE: If you see an empty screen for you candidate disks it is likely that
ASMLib has not been properly configured or installed. Try reconfigure them.
If you are sure that ASMLib has been properly configured click on 'Change
Discovery Path' and provide the correct destination.

ASM Password
>Specify and conform the password you want to use
>Next

Privileged OS Groups
>Assign the correct OS groups for OS authentication (mostly default is OK)
>Next

Installation Location
>ORACLE_BASE: /u01/app/oracle
 Software location: /u01/11.2.0/grid
>Next

Create Inventory
>Specify the locations: /u01/app/oraInventory
>Next

Perform Prerequisite Checks
>Check that status of all checks is Succeeded
>Next

Summary
>Finish

Execute Configuration Scripts
>Run the scripts as instructed in the screen
>OK

>Next

Message: The installation of the Grid Infrastructure was successfull.
>Close
```

3. Oracle Grid Infrastructure Patching

```
Apply patch set, if there is any.
```

4. Oracle Database 11g R2 Software Installation

```
# as oracle: copy DB software files into /u01/app/stage/ora11gr2db

# as root:
chmod -R 775 /u01/app/stage/ora11gr2db

-- start OUI
su - oracle
cd /u01/app/stage/ora11gr2db
./runInstaller

Follow the steps.
Install database software only.

When executing root.sh, select "y" for ovewriting questions.
```

5. Oracle Database 11g R2 Software Patching

```

```

6. Install EM Agent in cluster nodes (if required)

```

```

7. ASM Diskgroups Creation

```
# as grid user: start the ASM Configuration Assistant (ASMCA)
su - oracle
cd /u01/11.2.0/grid/bin
./asmca

>Disk Groups tab
>Create button
>Disk Group Name: FLASH
>Redundancy: External
>DISK2
>OK
>Exit
>Yes
```

8. Database Creation

```
# as oracle
cd /u01/app/oracle/product/11.2.0/db_1/bin
./dbca

follow the steps to create a database with sid: oradb

The DBCA may pop up the following message in the end of db creation:
"Error securing Database control, Database Control has been brought up in non-
secure mode. To secure the Database Control execute the following commands:
... "

# check Oracle processes:
ps -eo pid -o command | grep ora_ | grep -v grep
```

9. Postinstallation tasks

```
# backup the root.sh script
cp /u01/app/oracle/product/11.2.0/db_1/root.sh ~/root.sh.bak

cont>>
```

10. General Useful Postinstallation Tasks in Linux

- Consider using rlwrap utility with SQL*Plus and RMAN:
    - Using rlwrap Utility with RMAN in Unix-Based Systems
    - Using rlwrap Utility with SQL*Plus in Unix-Based Systems

```
/* Make crs_stat -t more readable */
/*  copy the following script into ~/scripts/crstat.sh */
#!/usr/bin/ksh
#
# Sample 10g CRS resource status query script
#
```

```
# Description:
#    - Returns formatted version of crs_stat -t, in tabular
#      format, with the complete rsc names and filtering keywords
#   - The argument, $RSC_KEY, is optional and if passed to the script, will
#     limit the output to HA resources whose names match $RSC_KEY.
# Requirements:
#   - $ORA_CRS_HOME should be set in your environment
# suggested scrip name: crstat.sh

RSC_KEY=$1
QSTAT=-u
AWK=/usr/bin/awk    # if not available use /usr/bin/awk

# Table header:echo ""
$AWK \
  'BEGIN {printf "%-45s %-10s %-18s\n", "HA Resource", "Target", "State";
          printf "%-45s %-10s %-18s\n", "-----------", "------", "-----";}'

# Table body:
$ORA_CRS_HOME/bin/crs_stat $QSTAT | $AWK \
 'BEGIN { FS="="; state = 0; }
  $1~/NAME/ && $2~/'$RSC_KEY'/ {appname = $2; state=1};
  state == 0 {next;}
  $1~/TARGET/ && state == 1 {apptarget = $2; state=2;}
  $1~/STATE/ && state == 2 {appstate = $2; state=3;}
  state == 3 {printf "%-45s %-10s %-18s\n", appname, apptarget, appstate;
state=0;}'


# then add the following in the .bashrc of oracle user
# if the file was saved in ~/scripts/crstat.sh
alias crstat='~/scripts/crstat.sh'
```

```
/* Easy Acces to crs and db homes */
# it is common to access bin directories in clusterware and db homes

# add the following to .bashrc of oracle user
alias db='cd /u01/app/oracle/product/11.2.0/db_1/bin'

# add the following to .bashrc of grid user
alias crs='cd /u01/app/oracle/crs/bin'
```

Managing Oracle Database Instance

Product Release Number

Oracle Database Release Number Format

```
SELECT * FROM PRODUCT_COMPONENT_VERSION;
SELECT * FROM V$VERSION
```

1.2.3.4.5

1 Major database release number
2 Maintenance release number
3 application server release number
4 component specific release number
5 platform specific release number

Obtaining License Information

```
-- Number of users and CPU/Processors
select L.SESSIONS_MAX, L.SESSIONS_WARNING, L.SESSIONS_CURRENT,
 L.SESSIONS_HIGHWATER,
 L.USERS_MAX, L.CPU_COUNT_CURRENT, L.CPU_SOCKET_COUNT_CURRENT,
 L.CPU_COUNT_HIGHWATER,
 L.CPU_CORE_COUNT_CURRENT, L.CPU_CORE_COUNT_HIGHWATER,
 L.CPU_SOCKET_COUNT_HIGHWATER
from v$license l;

-- Database Edition
select banner from v$version where BANNER like '%Edition%';

-- Oracle Partitioning installed
select decode(count(*), 0, 'No', 'Yes')
  from dba_part_tables
 where owner not in ('SYSMAN', 'SH', 'SYS', 'SYSTEM') and rownum = 1;

-- Oracle Spatial installed:
select decode(count(*), 0, 'No', 'Yes')
  from all_sdo_geom_metadata where rownum = 1;

-- Oracle RAC installed:
select decode(count(*), 0, 'No', 'Yes')
  from v$active_instances where rownum <= 2;
```

Managing the Instance Architecture

Obtaining Information about the Instance Processes

```
ps –ef | grep ora


select name, description from  V$BGPROCESS order by 1;
```

Obtaining Information about the SGA

```
select value from v$parameter where upper(name)='MEMORY_TARGET';
select value/1024/1024 mb
from v$parameter where upper(name) = 'SGA_MAX_SIZE'


-- size taken by a memory component
select pool, round(sum(BYTES)/1024/1024) MB
from V$SGASTAT
group by pool


select name , value/1024/1024 MB
from v$parameter
where upper(name) in (
'DB_CACHE_SIZE','DB_RECYCLE_CACHE_SIZE','DB_KEEP_CACHE_SIZE', '
DB_2k_CACHE_SIZE', 'DB_8k_CACHE_SIZE', 'DB_16k_CACHE_SIZE')


-- 10g
SELECT COMPONENT , CURRENT_SIZE/1024/1024 MB
FROM V$SGA_DYNAMIC_COMPONENTS
WHERE CURRENT_SIZE <>0


-- Oracle 11g
SELECT COMPONENT, ROUND(CURRENT_SIZE/1024/1024) CURRENT_SIZE ,
ROUND(MIN_SIZE/1024/1024) MIN, ROUND(MAX_SIZE/1024/1024) MAX
FROM V$MEMORY_DYNAMIC_COMPONENTS;
-- To know how Oracle has modified the memory area sizes by time
select START_TIME, END_TIME, STATUS, COMPONENT, OPER_TYPE, OPER_MODE,
PARAMETER, INITIAL_SIZE/1024/1024 INITIAL_SIZE_MB,
TARGET_SIZE/1024/1024 TARGET_SIZE_MB, FINAL_SIZE/1024/1024 FINAL_SIZE_MB
from V$MEMORY_RESIZE_OPS
order by START_TIME, END_TIME
```

Clearing the Buffer Cache

```
ALTER SYSTEM FLUSH SHARED_POOL;
ALTER SYSTEM FLUSH BUFFER_CACHE ;
```

Database Administration Authentication

Using Operating System Authentication

Following are the steps to create an Oracle administrator using OS authenticaion. Creating *normal* user who is capable of using OS authentication is a bit different. See External (OS) Authentication.

1. Create an operating system account for the user.
2. Add the account to the OSDBA or OSOPER operating system defined groups.

   Unix User Groups: dba, oper

   Windows: ORA_DBA, ORA_OPER
3. login to OS as the created user
4. In SQL*Plus

   CONNECT / AS SYSDBA

   CONNECT / AS SYSOPER

Using Password File Authentication

5. Create the password file
6. orapwd FILE=filename PASSWORD=password ENTRIES=max_users
7. alter system set REMOTE_LOGIN_PASSWORDFILE = EXCLUSIVE scope=SPFILE;
8. CONN / AS SYSDBA
9. GRANT SYSDBA TO GRANTEDUSER
10. CONN GRANTEDUSER/USERPASSWORD AS SYSDBA

Identifying Users SYSDBA or SYSOPER Users

```
SELECT * FROM V$PWFILE_USERS
```

Data Dictionary and Dynamic Performance Views

Data Dictionary Creation

`sql.bsq` creates base tables
`catalog.sql` creates data dictionary views
`catproc.sql` runs scripts required for server-side PL/SQL

- they run as SYSDBA
- they are located in \rdbms\admin

Startup and Shutdown

Startup Levels

```
STARTUP [FORCE] [RESTRICT] [PFILE=filename] [OPEN [RECOVER] [database] |MOUNT
|NOMOUNT]
ALTER DATABASE { MOUNT | OPEN }
ALTER DATABASE OPEN [READ WRITE| READ ONLY]
ALTER SYSTEM ENABLE | DISABLE RESTRICTED SESSION;
```

Shutdown Levels

```
SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT ]
```

Autostart of Database in Windows

- As a service or name OracleService*SID.* You can manage the service using ORADIM utility.
- You can control the shutdown type of the database when you shutdown its corresponding service by editing the registry entry ORA_<*SID*>_SHUTDOWNTYPE in Oracle home entry.

Automatically Starting Databases in Unix

- For Oracle 10.1, refer to Note 222813.1

```
/* Example 1 */
# compatible with Note 281912.1 (and Note 760051.1)
#(1) login as root
#(2) Set it to Y in /etc/oratab
mydb:/u01/app/oracle/product/10.2.0/db_1:Y
```

```
#(3) Create dbora script in the directory /etc/init.d
# fix ORA_HOME and ORA_OWNER values as required
vi /etc/init.d/dbora

#!/bin/bash
#
# description: Start/Stop the Databases..
#
# chkconfig: 2345 99 10
#
# processname: oracle
# config: /etc/oratab
# pidfile: /var/run/oracle.pid

# Source function library.
. /etc/init.d/functions

RETVAL=0
ORA_OWNER="oracle"
ORA_HOME="/u01/app/oracle/product/11.2.0/db_1"

prog="oracle"

start() {
 echo -n $"Starting $prog: "
 su - $ORA_OWNER -c "$ORA_HOME/bin/dbstart"
 su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl start"
 RETVAL=$?
 echo
 [ $RETVAL -eq 0 ] && touch /var/lock/subsys/dbora
 return $RETVAL
}

stop() {
 echo -n $"Stopping $prog: "
 su - $ORA_OWNER -c "$ORA_HOME/bin/dbshut"
 su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl stop"
 RETVAL=$?
 echo
 [ $RETVAL -eq 0 ] && rm -r /var/lock/subsys/dbora
 return $RETVAL
}

restart() {
 stop
 start
}
```

```
case "$1" in
start)
 start
 ;;
stop)
 stop
 ;;
restart)
 restart
 ;;
*)
echo $"Usage: $0 {start|stop|restart}"
exit 1
esac

exit $?


#(4) as root:
chgrp dba /etc/init.d/dbora
chmod 750 /etc/init.d/dbora

cd /sbin
chkconfig --add dbora
chkconfig --list | grep ora

# to test, restart or:
/etc/init.d/dbora start
```

```
/* Example 2 */
-- have been tested on an Oracle 10g Db on Linux 4
#(1) login as root
#(2) Set it to Y in /etc/oratab
mydb:/u01/app/oracle/product/10.2.0/db_1:Y

#(3) Create dbora script in the directory /etc/init.d
# fix ORACLE_HOME value as required

#!/bin/sh
#
# /etc/rc.d/init.d/dbora
# Description: Starts and stops the Oracle database and listeners
#
case "$1" in
  start)
        echo -n "Starting Oracle Databases: "
        echo "-------------------------------------" >> /var/log/oracle
        date +"! %T %a %D : Starting Oracle Databases as part of system up."
>> /var/log/oracle
        echo "-------------------------------------" >> /var/log/oracle
        su - oracle -c dbstart >> /var/log/oracle
```

```
        echo "Done."
        echo -n "Starting Oracle Listeners: "
        su - oracle -c "lsnrctl start" >> /var/log/oracle
        echo "Done."
        echo ""
        echo "------------------------------------" >> /var/log/oracle
        date +"! %T %a %D : Finished." >> /var/log/oracle
        echo "------------------------------------" >> /var/log/oracle
        touch /var/lock/subsys/oracle
        ;;
  stop)
        echo -n "Shutting Down Oracle Listeners: "
        echo "------------------------------------" >> /var/log/oracle
        date +"! %T %a %D : Shutting Down Oracle Databases as part of system
down." >> /var/log/oracle
        echo "------------------------------------" >> /var/log/oracle
        su - oracle -c "lsnrctl stop" >> /var/log/oracle
        echo "Done."
        rm -f /var/lock/subsys/oracle
        echo -n "Shutting Down Oracle Databases: "
        su - oracle -c dbshut >> /var/log/oracle
        echo "Done."
        echo ""
        echo "------------------------------------" >> /var/log/oracle
        date +"! %T %a %D : Finished." >> /var/log/oracle
        echo "------------------------------------" >> /var/log/oracle
        ;;
  restart)
        echo -n "Restarting Oracle Databases: "
        echo "------------------------------------" >> /var/log/oracle
        date +"! %T %a %D : Restarting Oracle Databases as part of system up."
>> /var/log/oracle
        echo "------------------------------------" >> /var/log/oracle
        su - oracle -c dbstop >> /var/log/oracle
        su - oracle -c dbstart >> /var/log/oracle
        echo "Done."
        echo -n "Restarting Oracle Listeners: "
        su - oracle -c "lsnrctl stop" >> /var/log/oracle
        su - oracle -c "lsnrctl start" >> /var/log/oracle
        echo "Done."
        echo ""
        echo "------------------------------------" >> /var/log/oracle
        date +"! %T %a %D : Finished." >> /var/log/oracle
        echo "------------------------------------" >> /var/log/oracle
        touch /var/lock/subsys/oracle
        ;;
  *)
        echo "Usage: oracle {start|stop|restart}"
        exit
esac

# (4) set file group and permissions
chgrp dba dbora
chmod 750 dbora

# (5) Create the symbolic links: In Linux,
ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/K01dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/S99dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/K01dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/S99dbora
```

```
/* Example 3 */
#(1) login as root
#(2) Set it to Y in /etc/oratab
mydb:/u01/app/oracle/product/10.2.0/db_1:Y

#(3) Create dbora script in the directory /etc/init.d
# fix ORACLE_HOME value as required
vi /etc/init.d/dbora

#!/bin/bash
#
# oracle Init file for starting and stopping
# Oracle Database. Script is valid for 10g and 11g versions.
#
# chkconfig: 35 80 30
# description: Oracle Database startup script

# Source function library.

. /etc/rc.d/init.d/functions

ORACLE_OWNER="oracle"
ORACLE_HOME="/u01/app/oracle/product/10.2.0/db_1"

case "$1" in
start)
 echo -n $"Starting Oracle DB:"
 su - $ORACLE_OWNER -c "$ORACLE_HOME/bin/dbstart $ORACLE_HOME"
 echo "OK"
 echo -n "Starting Oracle Listeners:"
 su - $ORACLE_OWNER -c "$ORACLE_HOME/bin/lsnrctl start"
 echo "OK"
;;
stop)
 echo -n $"Stopping Oracle DB:"
 su - $ORACLE_OWNER -c "$ORACLE_HOME/bin/dbshut $ORACLE_HOME"
 echo "OK"
 echo -n "Stopping Oracle Listeners:"
 su - $ORACLE_OWNER -c "$ORACLE_HOME/bin/lsnrctl stop"
 echo "OK"
;;
*)
echo $"Usage: $0 {start|stop}"
esac

# (4) set file group and permissions
chgrp dba /etc/init.d/dbora
chmod 750 /etc/init.d/dbora

# (5) Create the symbolic links: In Linux,
ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/K01dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/S99dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/K01dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/S99dbora
```

```
# (6) configuring execution for specific runlevels
chkconfig --add dbora --level 0356
```

Quiescing a Database

- Users will remain logged in, and they can continue to execute their requests that are in progress, while the database is in the quiesced state. The database, however, will block all new transactional requests except those made by the users SYS and SYSTEM (not DBA grantees).

```
ALTER SYSTEM QUIESCE RESTRICTED;
ALTER SYSTEM UNQUIESCE;
SELECT instance_name, status, shutdown_pending, active_state FROM v$instance;
```

Suspending a Database

- All reads from and writes to the datafiles and control files are suspended.

```
ALTER SYSTEM SUSPEND;
ALTER SYSTEM RESUME;
SELECT instance_name, status, shutdown_pending, active_state FROM v$instance;
```

Dropping a Database

- Datafiles, redo log files, and control files are removed automatically. Alert log and parameter file are not deleted.

```
CONNECT sys/sys_passwd AS SYSDBA
SHUTDOW IMMEDIATE
STARTUP RESTRICT MOUNT
SELECT name FROM v$database;
DROP DATABASE;
```

Initialization Files

Managing Initialization Files

- During startup, in $ORACLE_HOME/dbs (for UNIX/Linux) Oracle will look for the correct initialization file to use, in the following order:
  - spfile$ORACLE_SID.ora
  - spfile.ora
  - init$ORACLE_SID.ora

```
# currently used SPFiles (if null, pfile was used)
select * from v$parameter where name = 'spfile'
# create pfile
CREATE SPFILE='/u01/oracle/dbs/test_spfile.ora' FROM
PFILE='/u01/oracle/dbs/test_init.ora'
# use pecific spfile during startup
STARTUP PFILE = $ORACLE_HOME/dbs/initDBA1.ora
```

```
# PFILE can indicate to use SPFILE
SPFILE = /database/startup/spfileDBA1.ora
```

In Oracle 11g

```
# write current values of instance parameters
CREATE PFILE FROM MEMORY;
CREATE SPFILE FROM MEMORY;
```

Managing Parameters in SPFILE

```
-- display current value of a parameter
select * from v$parameter where name = 'spfile'
-- parameter values set in SPFILE
select NAME, ISSPECIFIED from V$SPPARAMETER where name like '%dest%';
-- in SQL Plus
show parameter target

-- set parameter value
alter system set parameter=value scope=SPFILE|MEMORY|BOTH

-- delete a parameter from SPFILe
ALTER SYSTEM RESET undo_suppress_errors SCOPE=BOTH SID='node01';

-- Only sessions started after the statement is executed are affected
-- this option is a must for the parameters whose ISSYS_MODIFIABLE column
--  in V$PARAMETER is DEFERRED. You cannot use it, if the column value is
FALSE
ALTER SYSTEM SET parameter_name DEFERRED;

-- undocumented parameters
SELECT
 a.ksppinm parameter,  a.ksppdesc description,
 b.ksppstvl session_value, c.ksppstvl instance_value
FROM x$ksppi a, x$ksppcv b, x$ksppsv c
WHERE
 a.indx = b.indx
 AND a.indx = c.indx
 AND SUBSTR (a.ksppinm,1,1) = '_'
ORDER BY a.ksppinm;
```

Alert and Trace Files

Monitoring Alert and Trace Files

```
# to obtain the Location of Trace Files
column NAME format a35
column VALUE format a35
select name, value from v$parameter
```

```
where name in ('background_dump_dest','user_dump_dest')

# to set maximum size of trace files (excluding the alert file)
#  in number of block unless you sepcify K or M
select * from v$parameter where upper(name )= 'MAX_DUMP_FILE_SIZE'
alter session set MAX_DUMP_FILE_SIZE='100M'
```

Managing Oracle Database Physical Structure

Managing Control Files

Obtaining Control File information

```
select value from v$parameter where upper(name)= 'CONTROL_FILES'
select * from v$controlfile;
select * from v$controlfile_record_section
```

Creating Additional Copies, Renaming, and Relocating Control Files

1. Shut down the database.

2. Copy an existing control file to a new location, using operating system commands.

3. Edit the CONTROL_FILES parameter
CONTROL_FILES="D:\ORACLE\ORADATA\ORA11G2\CONTROL01.CTL","D:\ORACLE\ORADATA\ORA1 1G2\CONTROL02.CTL";
4. Restart the database.

Note: if you are using SPFILE, STARTUP NOMOUNT then use ALTER SYSTEM SET .. SCOPE=SPFILE command.

Creating New Control Files

1. Make a list of all datafiles and redo log files of the database.
2. Shut down the database.
3. Back up all datafiles and redo log files of the database.
4. STARTUP NOMOUNT
5. Use the `CREATE CONTROLFILE` statement:

```
CREATE CONTROLFILE
SET DATABASE prod
LOGFILE GROUP 1 ('/u01/oracle/prod/redo01_01.log',
'/u01/oracle/prod/redo01_02.log'),
GROUP 2 ('/u01/oracle/prod/redo02_01.log',
'/u01/oracle/prod/redo02_02.log'),
GROUP 3 ('/u01/oracle/prod/redo03_01.log',
'/u01/oracle/prod/redo03_02.log')
RESETLOGS | NORESETLOGS
DATAFILE '/u01/oracle/prod/system01.dbf' SIZE 3M,
'/u01/oracle/prod/rbs01.dbs' SIZE 5M,
'/u01/oracle/prod/users01.dbs' SIZE 5M,
'/u01/oracle/prod/temp01.dbs' SIZE 5M
MAXLOGFILES 50
MAXLOGMEMBERS 3
MAXLOGHISTORY 400
MAXDATAFILES 200
MAXINSTANCES 6
ARCHIVELOG;
```

Specify the RESETLOGS clause if you have lost any redo log groups in addition to control files. In this case, you will need to recover from the loss of the redo logs (step 8). You must specify the RESETLOGS clause if you have renamed the database. Otherwise, select the NORESETLOGS clause.

Caution: The CREATE CONTROLFILE statement can potentially damage specified datafiles and redo log files. Omitting a filename can cause loss of the data in that file, or loss of access to the entire database.

6. Store a backup of the new control file on an offline storage device.
7. Edit the CONTROL_FILES initialization parameter
8. If you are renaming the database, edit the DB_NAME parameter in your instance parameter file.
9. Recover the database if necessary.
   - If the new control file was created using the NORESETLOGS clause, you can recover the database with complete, closed database recovery.
   - If the new control file was created using the RESETLOGS clause, you must specify USING BACKUP CONTROL FILE in your RECOVER command.

10. If you did not perform recovery, open the database normally.
ALTER DATABASE OPEN;
If you specified RESETLOGS when creating the control file:
ALTER DATABASE OPEN RESETLOGS;

Handling Errors During CREATE CONTROLFILE
If Oracle Database sends you an error (usually error ORA-01173, ORA-01176, ORA-01177, ORA-01215, or ORA-01216) when you attempt to mount and open the database after creating a new control file, the most likely cause is that you omitted a file from the CREATE CONTROLFILE statement or included one that should not have been listed.

Checking for Missing Files after Creating Control Files
Check the alert log to see if the database has detected inconsistencies between the data dictionary and the control file.

- If a datafile exists in the data dictionary but not in the new control file, the database creates a placeholder entry in the control file under the name MISSINGnnnn, where nnnn is the file number in decimal. MISSINGnnnn is flagged in the control file as being offline and requiring media recovery.
    - o If the actual datafile corresponding to MISSINGnnnn is read-only or offline normal, then you can make the datafile accessible by renaming MISSINGnnnn to the name of the actual datafile.
    - o If MISSINGnnnn corresponds to a datafile that was not read-only or offline normal, you must drop the tablespace containing the datafile.

Backing Up Control Files

```
-- copy of the control file:
ALTER DATABASE BACKUP CONTROLFILE TO '/oracle/backup/control.bkp';

-- commands to re-create the control file:
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
Show parameter user_dump_dest
```

Manage the Size of Control Files

It is affected by MAXDATAFILES, MAXLOGFILES, MAXLOGMEMBERS, MAXLOGHISTORY, and MAXINSTANCES parameters in the CREATE DATABASE statement. Also it is affected by CONTROL_FILE_RECORD_KEEP_TIME

Multiplexing the Control File

1. Alter the SPFILE:
```
ALTER SYSTEM SET control_files
 ='$HOME/ORADATA/u01/ctrl01.ctl','$HOME/ORADATA/u02/ctrl02.ctl' SCOPE=SPFILE;
```
2. Shut down the database
3. Create additional control files:
```
cp $HOME/ORADATA/u01/ctrl01.ctl $HOME/ORADATA/u02/ctrl02.ctl
```
4. Start the database:
```
startup
```

Maintaining Online Redo Log Files

Forcing Log Switches and Checkpoints

```
# Forcing a log switch
ALTER SYSTEM SWITCH LOGFILE;
# forcing checkpoints in seconds
FAST_START_MTTR_TARGET = 600
ALTER SYSTEM CHECKPOINT;
```

Adding Online Redo Log File Groups

```
# GROUP n is optional
ALTER DATABASE ADD LOGFILE GROUP 3
('/ORADATA/u01/log3a.rdo', '/ORADATA/u02/log3b.rdo')
SIZE 1M;
```

Adding Online Redo Log File Members

```
ALTER DATABASE ADD LOGFILE MEMBER
'/ORADATA/u04/log1c.rdo' TO GROUP 1,
'/ORADATA/u04/log2c.rdo' TO GROUP 2,
'/ORADATA/u04/log3c.rdo' TO GROUP 3;
```

Dropping Online Redo Log File Groups

```
SELECT GROUP#, ARCHIVED, STATUS FROM V$LOG;
ALTER SYSTEM SWITCH LOGFILE;
ALTER DATABASE DROP LOGFILE GROUP 3;
```

Dropping Online Redo Log File Members

```
ALTER DATABASE DROP LOGFILE MEMBER '$HOME/ORADATA/u04/log3c.rdo';
```

Relocating and Renaming Redo Log Members

```
SHUTDOWN
Copy the redo log files to the new location.
STARTUP MOUNT
```

```
ALTER DATABASE
RENAME FILE '/diska/logs/log1a.rdo', '/diska/logs/log2a.rdo'
TO '/diskc/logs/log1c.rdo', '/diskc/logs/log2c.rdo';
ALTER DATABASE OPEN;
```

Verifying Blocks in Redo Log Files

```
# it defaults to TURE
alter system set DB_BLOCK_CHECKSUM=true ;
```

Clearing a Redo Log File

```
# if DB stops becuase log file is corrupted
ALTER DATABASE CLEAR LOGFILE GROUP 2;
# the redo log file will be available even if not archived
ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 2;
```

Viewing Redo Log Information

```
SELECT * FROM V$LOG;
# STATUS: INVALID (inaccessible), STALE (incomplete), DELETED, Blank (in use)
SELECT * FROM V$LOGFILE;
SELECT * FROM V$LOG G, V$LOGFILE M where G.GROUP#=M.GROUP# order by M.GROUP#
```

Managing Archived Redo Logs

Obtaining Information about Archive Log

```
-- in  SQL*Plus
ARCHIVE LOG LIST
-- to know archivelog mode of the database
select LOG_MODE from V$DATABASE

-- historical archived log information from the control file
select
 RECID,
 NAME,
 DEST_ID,
 THREAD#,
 SEQUENCE#,
 RESETLOGS_CHANGE#,
 RESETLOGS_TIME,
 RESETLOGS_ID,
 FIRST_CHANGE#,
 FIRST_TIME,
 NEXT_CHANGE#,
 NEXT_TIME,
 BLOCKS * BLOCK_SIZE/1024/1024 MB ,
 CREATOR,
 ARCHIVED,
 DELETED,
 STATUS,
 COMPLETION_TIME,
```

```
 END_OF_REDO,
 BACKUP_COUNT,
 ARCHIVAL_THREAD#,
 IS_RECOVERY_DEST_FILE,
 COMPRESSED,
 FAL,
 BACKED_BY_VSS
from V$ARCHIVED_LOG;

-- information about archive log destinations and their status
select * from V$ARCHIVE_DEST

-- information about ARCn processes
select * from V$ARCHIVE_PROCESSES

-- information about any backup made on archived log files
select * from V$BACKUP_REDOLOG

-- online groups and which one to be archived
select * from V$LOG

-- log history information
select * from V$LOG_HISTORY
```

Changing the Database Archiving Mode

```
select log_mode from v$database ;

CONN / AS SYSDBA
SHUTDOWN
Back up the database
see Specifying Archive Destinations ( next section )
STARTUP MOUNT
ALTER DATABASE ARCHIVELOG; -- or NOARCHIVELOG
ALTER DATABASE OPEN;
SHUTDOWN IMMEDIATE
Back up the database
```

Specifying Archive Destinations and their Options

```
-- to local destinations
LOG_ARCHIVE_DEST_1 = 'LOCATION=/disk1/archive'
LOG_ARCHIVE_DEST_1 = 'LOCATION=USE_DB_RECOVERY_FILE_DEST'

-- default is OPTIONAL, REOPEN in seconds (default 300)
--  if REOPEN is omitted, ARCn will never open a destination after a failure
LOG_ARCHIVE_DEST_2 = 'LOCATION=/disk2/archive MANDATORY REOPEN=600'
LOG_ARCHIVE_DEST_3 = 'LOCATION=/disk3/archive OPTIONAL'

-- to a standby db
LOG_ARCHIVE_DEST_4 = 'SERVICE = standby1'

-- control file format
```

```
LOG_ARCHIVE_FORMAT=t%t_s%s_r%r.arc
```

Specifying the Minimum Number of Successful Destinations

```
alter system set LOG_ARCHIVE_MIN_SUCCEED_DEST=1
```

Controlling Archiving to a Destination

```
alter system set LOG_ARCHIVE_DEST_STATE_2 = DEFER
alter system set LOG_ARCHIVE_DEST_STATE_2 = ENABLE
```

Controlling Trace Output Generated by the Archivelog Process

LOG_ARCHIVE_TRACE takes combination of:

0	Disable archivelog tracing. This is the default.
1	Track archival of redo log file.
2	Track archival status for each archivelog destination.
4	Track archival operational phase.
8	Track archivelog destination activity.
16	Track detailed archivelog destination activity.
32	Track archivelog destination parameter modifications.
64	Track ARCn process state activity.
128	Track FAL (fetch archived log) server related activities.
256	Supported in a future release.
512	Tracks asynchronous LGWR activity.
1024	RFS physical client tracking.
2048	ARCn/RFS heartbeat tracking.
4096	Track real-time apply

```
--  LOG_ARCHIVE_TRACE defaults to 0
select value from v$parameter where upper(name)='LOG_ARCHIVE_TRACE';
-- database must be mounted but not open.
ALTER SYSTEM SET LOG_ARCHIVE_TRACE=12;
```

Managing Tablespaces

Obtaining Tablespace Information

```
-- tablespace size usage report (for large db (>100GB), it takes long time)
BREAK ON REPORT
COMPUTE SUM OF tbsp_size ON REPORT
compute SUM OF used ON REPORT
compute SUM OF free ON REPORT
COL tbspname FORMAT a20 HEADING 'Tablespace Name'
COL tbsp_size FORMAT 999,999 HEADING 'Size|(MB)'
COL used FORMAT 999,999 HEADING 'Used|(MB)'
COL free FORMAT 999,999 HEADING 'Free|(MB)'
COL pct_used FORMAT 999 HEADING'% Used'

SELECT df.tablespace_name tbspname,
 sum(df.bytes)/1024/1024 tbsp_size,
 nvl(sum(e.used_bytes)/1024/1024,0) used,
 nvl(sum(f.free_bytes)/1024/1024,0) free,
 nvl((sum(e.used_bytes)*100)/sum(df.bytes),0) pct_used
FROM DBA_DATA_FILES df,
```

```
 (SELECT file_id,
  SUM(nvl(bytes,0)) used_bytes
  FROM dba_extents
  GROUP BY file_id) e,
 (SELECT MAX(bytes) free_bytes, file_id
  FROM dba_free_space
  GROUP BY file_id) f
WHERE e.file_id(+) = df.file_id
  AND df.file_id = f.file_id(+)
  GROUP BY df.tablespace_name
ORDER BY 5 DESC

-- for a specific tablespace:
SELECT df.tablespace_name tbspname,
 sum(df.bytes)/1024/1024 tbsp_size,
 nvl(sum(e.used_bytes)/1024/1024,0) used,
 nvl(sum(f.free_bytes)/1024/1024,0) free,
 nvl((sum(e.used_bytes)*100)/sum(df.bytes),0) pct_used
FROM DBA_DATA_FILES df,
 (SELECT file_id,
  SUM(nvl(bytes,0)) used_bytes
  FROM dba_extents
  WHERE TABLESPACE_NAME='REC_DATA'
  GROUP BY file_id) e,
 (SELECT MAX(bytes) free_bytes, file_id
  FROM dba_free_space
  WHERE TABLESPACE_NAME='REC_DATA'
  GROUP BY file_id) f
WHERE e.file_id(+) = df.file_id
  AND df.file_id = f.file_id(+)
  AND TABLESPACE_NAME='REC_DATA'
  GROUP BY df.tablespace_name;
-- ************************************************************


-- tablespace sizes (without getting used and free space) (quick response):
SELECT DF.TABLESPACE_NAME TBSPNAME,
  ROUND(SUM(DF.BYTES)/1024/1024/1024,2) GB,
  COUNT(FILE_NAME) DATAFILES
 FROM DBA_DATA_FILES DF
 GROUP BY ROLLUP(DF.TABLESPACE_NAME )
 ORDER BY 1



-- free space in temp tablespaces:
select sum(free_blocks)
from gv$sort_segment
where tablespace_name = 'USER_TEMP' ;



-- tablespace info from control file
select TS#,NAME,INCLUDED_IN_DATABASE_BACKUP,
       BIGFILE,FLASHBACK_ON,ENCRYPT_IN_BACKUP
from   V$TABLESPACE;


-- descriptions of tablespaces
select TABLESPACE_NAME,BLOCK_SIZE,INITIAL_EXTENT,
  NEXT_EXTENT NEXT_EXTENT_SIZE,MIN_EXTENTS,MAX_EXTENTS,
```

```
  PCT_INCREASE,MIN_EXTLEN,STATUS,
  CONTENTS,LOGGING,FORCE_LOGGING,
  EXTENT_MANAGEMENT,ALLOCATION_TYPE,PLUGGED_IN,
  SEGMENT_SPACE_MANAGEMENT,DEF_TAB_COMPRESSION,RETENTION
from DBA_TABLESPACES


-- segments contained in tablespaces
select OWNER,SEGMENT_NAME,PARTITION_NAME,SEGMENT_TYPE,S.TABLESPACE_NAME,
  HEADER_FILE,HEADER_BLOCK,S.BYTES/1024/1024 SEGMENT_SIZE_MB,
  D.BLOCKS SEGMENT_BLOCKS,EXTENTS,S.INITIAL_EXTENT,S.NEXT_EXTENT
NEXT_EXTENT_SIZE, S.MIN_EXTENTS,S.MAX_EXTENTS,S.PCT_INCREASE,
  FREELISTS,FREELIST_GROUPS,D.RELATIVE_FNO, D.FILE_NAME
from DBA_SEGMENTS S, DBA_TABLESPACES T, DBA_DATA_FILES D
where S.TABLESPACE_NAME = T.TABLESPACE_NAME AND T.TABLESPACE_NAME =
      D.TABLESPACE_NAME
  AND S.RELATIVE_FNO = D.RELATIVE_FNO
  AND T.TABLESPACE_NAME NOT IN ('SYSAUX','SYSTEM')
order BY S.TABLESPACE_NAME

-- extents contained by tablespaces
SELECT EXTENT_ID, E.BLOCK_ID,E.BYTES/1024 EXTENT_SIZE_KB,
E.OWNER,E.SEGMENT_NAME,E.PARTITION_NAME,
  E.SEGMENT_TYPE,D.FILE_NAME, S.TABLESPACE_NAME,HEADER_FILE,
  HEADER_BLOCK SEG_HEADER_BLOCK,S.BYTES/1024/1024 SEGMENT_SIZE_MB,D.BLOCKS
SEGMENT_BLOCKS,  EXTENTS SEG_EXTENTS
FROM DBA_EXTENTS E, DBA_SEGMENTS S, DBA_DATA_FILES D
WHERE E.OWNER=S.OWNER AND E.SEGMENT_NAME = S.SEGMENT_NAME
  AND NVL(E.PARTITION_NAME,'0') = NVL(S.PARTITION_NAME,'0')
  AND E.SEGMENT_TYPE = S.SEGMENT_TYPE
  AND E.FILE_ID = D.FILE_ID
  AND S.TABLESPACE_NAME NOT IN ('SYSAUX','SYSTEM')
ORDER BY E.SEGMENT_NAME, E.OWNER, E.PARTITION_NAME,E.EXTENT_ID


-- free extents within tablespaces
SELECT F.TABLESPACE_NAME,F.FILE_ID,F.BLOCK_ID,
  F.BYTES/1024/1024 FREE_MB, D.FILE_NAME
FROM DBA_FREE_SPACE F, DBA_DATA_FILES D
WHERE F.FILE_ID = D.FILE_ID
UNION
SELECT F.TABLESPACE_NAME, TO_NUMBER('') AS FILE_ID, TO_NUMBER('') AS BLOCK_ID,
  SUM(F.BYTES/1024/1024) FREE_MB, TO_CHAR('') AS FILE_NAME
FROM DBA_FREE_SPACE F
GROUP BY F.TABLESPACE_NAME, TO_NUMBER('') , TO_NUMBER('') ,TO_CHAR('')
ORDER BY TABLESPACE_NAME


-- data files (from control file)
SELECT FILE#,T.NAME TABLESPACE_NAME,D.NAME FILENAME,
 CREATION_CHANGE#,CREATION_TIME, RFILE#,STATUS,
 ENABLED,CHECKPOINT_CHANGE#,CHECKPOINT_TIME,
 UNRECOVERABLE_CHANGE#,UNRECOVERABLE_TIME,LAST_CHANGE#,
 LAST_TIME,OFFLINE_CHANGE#,ONLINE_CHANGE#,
 ONLINE_TIME,BYTES/1024/1024 FILESIZE_MB,BLOCKS,
 CREATE_BYTES,BLOCK_SIZE,
 PLUGGED_IN,BLOCK1_OFFSET,AUX_NAME
FROM V$DATAFILE D, V$TABLESPACE T
WHERE D.TS# = T.TS#
ORDER BY TABLESPACE_NAME, D.RFILE#


-- data files
```

```
select FILE_NAME,FILE_ID,T.TABLESPACE_NAME,
 F.STATUS FILE_STATUS,
 RELATIVE_FNO,
 AUTOEXTENSIBLE,
 ROUND(BYTES/1024/1024,2) FILESIZE_MB,
 ROUND(MAXBYTES/1024/1024,2) MAXSIZE_MB,
 ROUND((INCREMENT_BY*T.BLOCK_SIZE)/1024/1024,2) AUTOEXTENSION_SIZE_MB,
 ROUND(USER_BYTES/1024/1024,2) AVAILABLE_FOR_DATA_MB -- remaining size used
from DBA_DATA_FILES F, DBA_TABLESPACES T             --  for storing metadata
where F.TABLESPACE_NAME = T.TABLESPACE_NAME
order BY TABLESPACE_NAME, F.RELATIVE_FNO


-- tempfiles included in tablespaces
select M.NAME TEMPFILE,FILE#,T.NAME TABLESPACE_NAME,
 CREATION_TIME,M.TS#,RFILE#,
 STATUS,ENABLED,BYTES,
 BLOCKS,CREATE_BYTES,BLOCK_SIZE
from   V$TEMPFILE M, V$TABLESPACE T
where  M.TS# = T.TS#
order by T.NAME

select FILE_NAME,FILE_ID,T.TABLESPACE_NAME,
 F.STATUS FILE_STATUS,
 RELATIVE_FNO,
 AUTOEXTENSIBLE,
 ROUND(BYTES/1024/1024,2) FILESIZE_MB,
 ROUND(MAXBYTES/1024/1024,2) MAXSIZE_MB,
 ROUND((INCREMENT_BY*T.BLOCK_SIZE)/1024/1024,2) AUTOEXTENSION_SIZE_MB
from DBA_TEMP_FILES F, DBA_TABLESPACES
where F.TABLESPACE_NAME = T.TABLESPACE_NAME
order BY TABLESPACE_NAME, F.RELATIVE_FNO


-- extents in all locally managed temporary tablespaces
SELECT E.TABLESPACE_NAME,E.FILE_ID,
 BLOCK_ID BEGIN_BLOCK#,
 ROUND(E.BYTES/1024,2) EXTENT_SIZE_KB,
 E.BLOCKS,OWNER
FROM  V$TEMP_EXTENT_MAP E


-- tablespace groups
select GROUP_NAME, TABLESPACE_NAME
from DBA_TABLESPACE_GROUPS
order BY TABLESPACE_NAME


-- user qoutas
SELECT
 USERNAME,TABLESPACE_NAME, BYTES/1024 SIZE_KB,
 DECODE(MAX_BYTES,-1,-1,MAX_BYTES/1024/1024) MAX_MB
FROM DBA_TS_QUOTAS
ORDER BY USERNAME


-- diskspace usage by USER
select owner, round(sum(bytes)/1024/1024,2) space_in_mb
from dba_segments
group by owner
WHERE OWNER NOT IN ('SYSTEM', 'SYSMAN',
'SYS','WKSYS','WK_TEST','WMSYS','XDB','OUTLN','PERFSTAT','OLAPSYS','ORDSYS','M
DSYS','EXFSYS','DMSYS','CTXSYS','REPTEST','SCOTT','RMAN')
```

```
order by round(sum(bytes)/1024/1024,2) desc;


-- user temporary usage
-- SEGTYPE: SORT,HASH,DATA,INDEX,LOB_DATA,LOB_INDEX
SELECT USERNAME,SESSION_NUM SESSION_SN,
SQLADDR,SQLHASH,SQL_ID,
TABLESPACE,
SEGTYPE,SEGFILE# INIT_EXTENT_FILE#,SEGBLK#
INIT_EXTENT_BLK#,EXTENTS,BLOCKS,SEGRFNO#
FROM V$TEMPSEG_USAGE
```

Creating a Locally Managed Tablespace

```
CREATE TABLESPACE lmtbsb DATAFILE '/u02/oracle/data/lmtbsb01.dbf' SIZE 50M
EXTENT MANAGEMENT AUTOALLOCATE;
CREATE TABLESPACE lmtbsb DATAFILE '/u02/oracle/data/lmtbsb01.dbf' SIZE 50M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K; -- default 1MB


-- creating a tablespace in an ASM diskgroup
CREATE TABLESPACE sample DATAFILE '+dgroup1';
CREATE TABLESPACE satbs DATAFILE '+DATA' size 50m;
```

Specifying Segment Space Management

```
CREATE TABLESPACE lmtbsb DATAFILE '/u02/oracle/data/lmtbsb01.dbf' SIZE 50M
EXTENT MANAGEMENT LOCAL
 SEGMENT SPACE MANAGEMENT AUTO; -- though it's the default
  -- the other option is MANUAL
```

Adding Space to Tablespace

```
ALTER TABLESPACE test01 ADD DATAFILE '..'  SIZE 1000M;
ALTER DATABASE DATAFILE '…' RESIZE 500m;
```

Specifying Nonstandard Block Sizes for Tablespaces

```
-- must be 2KB, 4KB, 8KB, 16KB, or 32KB.
-- requirement
set DB_CACHE_SIZE, DB_nK_CACHE_SIZE
select value/1024 KB from v$parameter where name='db_block_size';
CREATE TABLESPACE lmtbsb DATAFILE '/u02/oracle/data/lmtbsb01.dbf' SIZE 50M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K BLOCKSIZE 8K;
```

Using Bigfile Tablespace (BFT)

- Oracle recommends that you change the extent allocation policy from AUTOALLOCATE, which is the default, to UNIFORM and set a very high extent size.

```
select TABLESPACE_NAME, BIGFILE from DBA_TABLESPACES order by BIGFILE;
-- use BigFile tablespaces only with ASM or RAID volume manager.
-- it can take up to 4G blocks. with 8K blocks = 32 terabyte datafile
CREATE BIGFILE TABLESPACE bigtbs -- the other option SMALLFILE
DATAFILE '/u02/oracle/data/bigtbs01.dbf' SIZE 50G -- T is acceptable
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 65536K; -- default 1m
ALTER TABLESPACE bigtbs RESIZE 80G;
```

```
ALTER TABLESPACE bigtbs AUTOEXTEND ON NEXT 20G;
-- with this sitting, by default all tbs created later will be BFT
CREATE DATABASE SET DEFAULT BIGFILE tablespace … -- the other option SMALLFILE
ALTER TABLESPACE SET DEFAULT BIGFILE TABLESPACE;
SELECT property_value FROM database_properties
WHERE  property_name='DEFAULT_TBS_TYPE';
```

Using Temporary Tablespace

- Oracle recommends creating temporary tablespaces with multiples-of-64KB extent sizes. For large data warehousing, make it 1MB.

```
select * from V$TEMPFILE;
select * from DBA_TEMP_FILES;


CREATE TEMPORARY TABLESPACE lmtemp TEMPFILE '/u02/oracle/data/lmtemp01.dbf'
 SIZE 20M REUSE
 EXTENT MANAGEMENT LOCAL UNIFORM SIZE 16M; -- default is 1M (AUTOALLOCATE not
  allowed)


CREATE BIGFILE TABLESPACE bigtbs
TEMPFILE '/u02/oracle/data/bigtbs01.dbf' SIZE 50G
ALTER TABLESPACE lmtemp ADD TEMPFILE '/u02/oracle/data/lmtemp02.dbf' SIZE 18M
REUSE;


-- doesn't apply on default temporary tablespace
ALTER TABLESPACE lmtemp TEMPFILE OFFLINE;
ALTER TABLESPACE lmtemp TEMPFILE ONLINE;
ALTER DATABASE TEMPFILE '/u02/oracle/data/lmtemp02.dbf' OFFLINE;
ALTER DATABASE TEMPFILE '/u02/oracle/data/lmtemp02.dbf' ONLINE;
ALTER DATABASE TEMPFILE '/u02/oracle/data/lmtemp02.dbf' RESIZE 18M;


-- the tablespace isn't dropped
ALTER DATABASE TEMPFILE '/u02/oracle/data/lmtemp02.dbf' DROP INCLUDING
DATAFILES;
```

Renaming a Tempfile

```
ALTER DATABASE TEMPFILE 'C:\ORACLE\ORADATA\ORCL\TEMP02.DBF' OFFLINE;
ren C:\ORACLE\ORADATA\ORCL\TEMP02.DBF TEMP03.DBF
ALTER DATABASE RENAME FILE 'C:\ORACLE\ORADATA\ORCL\TEMP02.DBF' TO
'C:\ORACLE\ORADATA\ORCL\TEMP03.DBF';
ALTER DATABASE TEMPFILE 'C:\ORACLE\ORADATA\ORCL\TEMP03.DBF' ONLINE;
```

Shrinking Temporary Tablespace

```
ALTER TABLESPACE temp SHRINK SPACE KEEP 1000M;
ALTER TABLESPACE temp SHRINK SPACE TEMPFILE tempfile
'/u01/app/oracle/oradata/prod1/temp02.dbf' KEEP 100m;
SELECT file#, name, bytes/1024/1024 mb FROM v$tempfile;
```

Using Default Temporary Tablespace

```
ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temptbs02; -- can be temp tbs grp
SELECT PROPERTY_NAME, PROPERTY_VALUE
FROM database_properties
WHERE property_name='DEFAULT_TEMP_TABLESPACE';
```

Using Temporary Tablespace Groups

```
CREATE TEMPORARY TABLESPACE lmtemp2 TEMPFILE '/u02/oracle/data/lmtemp201.dbf'
SIZE 50M TABLESPACE GROUP group1;

ALTER TABLESPACE lmtemp TABLESPACE GROUP group2;

-- remove it from a group
ALTER TABLESPACE lmtemp3 TABLESPACE GROUP '';

-- Assigning a Tablespace Group as the Default Temporary Tablespace
ALTER DATABASE mydb DEFAULT TEMPORARY TABLESPACE group2;

select GROUP_NAME, TABLESPACE_NAME
from DBA_TABLESPACE_GROUPS
order BY TABLESPACE_NAME
```

Suppressing Redo Generation for a Tablespace

```
CREATE TABLESPACE .. NOLOGGING;
```

Controlling Tablespaces Availability

```
-- NORMAL, TEMPORARY, IMMEDIATE (not possible in NOARCHIVELOG)
ALTER TABLESPACE users OFFLINE NORMAL;

-- media recovery required if it was offline using TEMPORARY or IMMEDIATE
ALTER TABLESPACE users ONLINE;
```

Using Read-Only Tablespaces

```
-- backup the tablespace after making it read only
-- it waits for all transactions started before
ALTER TABLESPACE flights READ ONLY;

-- to list blocking transactions:
SELECT SQL_TEXT, SADDR
FROM V$SQLAREA,V$SESSION
WHERE V$SQLAREA.ADDRESS = V$SESSION.SQL_ADDRESS
AND SQL_TEXT LIKE 'alter tablespace%';

-- all transactions on top of SADDR retuned by previous statement
--  are blocking transactions

SELECT T.SES_ADDR, T.START_SCNB, S.USERNAME, S.MACHINE

FROM V$TRANSACTION T, V$SESSION S
WHERE T.SES_ADDR = S.SADDR
ORDER BY START_SCNB;

-- back to read write
ALTER TABLESPACE flights READ WRITE;
```

Renaming Tablespaces

```
ALTER TABLESPACE users RENAME TO usersts;
```

Default Permanent Tabelspace

```
SELECT property_value FROM database_properties WHERE
```

```
 property_name='DEFAULT_PERMANENT_TABLESPACE';
ALTER DATABASE DEFAULT TABLESPACE users;
```

Dropping Tablespaces

```
DROP TABLESPACE users INCLUDING CONTENTS;
DROP TABLESPACE test01 CASCADE CONSTRAINTS;
DROP TABLESPACE users INCLUDING CONTENTS CASCADE CONSTRAINTS;
DROP TABLESPACE users INCLUDING CONTENTS AND DATAFILES;
```

Managing the SYSAUX Tablespace

A typical system with an average of 30 concurrent active sessions may require approximately 200 to 300 MB of space for its AWR data.

```
-- to monitor is occupants
--  to know which procedure to use to move an occupant
SELECT OCCUPANT_NAME , OCCUPANT_DESC , SCHEMA_NAME ,MOVE_PROCEDURE
,MOVE_PROCEDURE_DESC ,SPACE_USAGE_KBYTES FROM V$SYSAUX_OCCUPANTS
```

Diagnosing and Repairing Locally Managed Tablespace Problems

Verifying the Integrity of Segments Created in ASSM Tablespaces.

- Use DBMS_SPACE_ADMIN.ASSM_SEGMENT_VERIFY.
- If ASSM is disabled, use SEGMENT_VERIFY.
- After execution, check sid_ora_process_ID.trc in USER_DUMP_DEST.
- The parameter verify_option takes one of the following constants:
  - SEGMENT_VERIFY_DEEP (9)
  - SEGMENT_VERIFY_BASIC (10) default
  - SEGMENT_VERIFY_SPECIFIC (11) then the attrib parameter is considered.
- attrib takes one of the following constants:
  - HWM_CHECK (12) checks whether high water mark information is accurate.
  - BMB_CHECK (13) checks whether space bitmap blocks have correct backpointers to the segment header.
  - SEG_DICT_CHECK (14) checks whether dictionary information for segment is accurate.
  - EXTENT_TS_BITMAP_CHECK (15) checks whether extent maps are consistent with file level bitmaps.
  - DB_BACKPOINTER_CHECK (16) checks whether datablocks have correct backpointers to the space metadata blocks.
  - EXTENT_SEGMENT_BITMAP_CHECK (17) checks whether extent map in the segment match with the bitnaps in the segment.
  - BITMAPS_CHECK (18) checks whether space bitmap blocks are accurate.

```
declare
 v_segname  varchar2(100) := 'EMPLOYEES';
 v_segowner varchar2(100) :='HR';
 v_segtype  varchar2(100) :='TABLE';
 v_tbs      varchar2(100);
```

```
begin

 select tablespace_name
   into v_tbs
  from  dba_segments
 where  segment_name=v_segname and owner=v_segowner
        and segment_type=v_segtype;

DBMS_SPACE_ADMIN.ASSM_SEGMENT_VERIFY (
 segment_owner  =>v_segowner,
 segment_name   =>v_segname,
 segment_type   =>v_segtype,
 partition_name =>'',
 verify_option  => DBMS_SPACE_ADMIN.SEGMENT_VERIFY_DEEP
);
end;
/


-- Check sid_ora_process_ID.trc in USER_DUMP_DEST
select value from v$parameter where upper(name)='USER_DUMP_DEST'
```

```
declare
 v_segname  varchar2(100) := 'EMPLOYEES';
 v_segowner varchar2(100) :='HR';
 v_segtype  varchar2(100) :='TABLE';
 v_tbs      varchar2(100);
begin

 select tablespace_name
   into v_tbs
  from  dba_segments
 where  segment_name=v_segname and owner=v_segowner
        and segment_type=v_segtype;

DBMS_SPACE_ADMIN.ASSM_SEGMENT_VERIFY (
 segment_owner  =>v_segowner,
 segment_name   =>v_segname,
 segment_type   =>v_segtype,
 partition_name =>'',
 verify_option  => DBMS_SPACE_ADMIN.SEGMENT_VERIFY_SPECIFIC,
 attrib         => DBMS_SPACE_ADMIN.BMB_CHECK
);
end;
/


-- Check sid_ora_process_ID.trc in USER_DUMP_DEST
select value from v$parameter where upper(name)='USER_DUMP_DEST'
```

Checking Consistency of Segment Extent Map with Tablespace File Bitmaps

- Use ASSM_SEGMENT_VERIFY to segment residing in a tablespace with automatic segment space management enabled and SEGMENT_VERIFY when it is disabled.
- After execution, check sid_ora_process_ID.trc in USER_DUMP_DEST.

```
conn sys as sysdba
declare
 v_segname varchar2(100) := 'EMPLOYEES';
 v_segowner varchar2(100) :='HR';
```

```
 v_tbs varchar2(100);
 v_fno number ;
 v_rfno number;
 v_hdr number;
begin
 -- retreive tablespace name, absolute file number
 select tablespace_name, header_file, header_block
   into v_tbs, v_fno, v_hdr
  from  dba_segments
 where  segment_name=v_segname and owner=v_segowner;

 select relative_fno
   into v_rfno
   from dba_data_files  where tablespace_name = v_tbs and file_id=v_fno;

DBMS_SPACE_ADMIN.SEGMENT_VERIFY(
 tablespace_name      =>v_tbs,
 header_relative_file =>v_rfno,
 header_block         =>v_hdr,
 verify_option        =>dbms_space_admin.SEGMENT_VERIFY_EXTENTS_GLOBAL --
default SEGMENT_VERIFY_EXTENTS
 );
end;
/
-- Check sid_ora_process_ID.trc in USER_DUMP_DEST
select value from v$parameter where upper(name)='USER_DUMP_DEST'
```

Verifying the Integrity of ASSM Tablespaces

- Use DBMS_SPACE_ADMIN.ASSM_TABLESPACE_VERIFY
- If ASSM is disabled, use TABLESPACE_VERIFY
- After execution, check sid_ora_process_ID.trc in USER_DUMP_DEST.
- The parameter ts_option takes one of the following constants:
  - TS_VERIFY_BITMAPS (19) (Default) The bitmaps are verified against the extents. This will detect bits that are marked used or free wrongly and will also detect multiple allocation of extents. The file metadata will be validated against file$ and control file.
  - TS_VERIFY_DEEP (20) verifies the file bitmaps as well perform checks on all the segments.
  - TS_SEGMENTS (21) This option is used to invoke SEGMENT_VERIFY on all the segments in the tablespace.
- segment_option: when the TABLESPACE_VERIFY_SEGMENTS or TABLESPACE_VERIFY_DEEP is selected, the SEGMENT_OPTION can be specified optionally. When TS_VERIFY_SEGMENTS is specified, segment_option can be one of the following:
  - SEGMENT_VERIFY_BASIC (9)
  - SEGMENT_VERIFY_DEEP (10)
- The value of segment_option is NULL when TS_VERIFY_DEEP or TS_VERIFY_BASIC is specified.
- After execution, check sid_ora_process_ID.trc in USER_DUMP_DEST

```
conn sys as sysdba

select name from v$tablespace order by name;

begin
```

```
DBMS_SPACE_ADMIN.ASSM_TABLESPACE_VERIFY(
 tablespace_name =>'EXAMPLE',
 ts_option => DBMS_SPACE_ADMIN.TS_VERIFY_DEEP,
 segment_option =>NULL);
end;
/

-- Check sid_ora_process_ID.trc in USER_DUMP_DEST
select value from v$parameter where upper(name)='USER_DUMP_DEST'
```

Marking the Segment Corrupt or Valid

- corrupt_option takes one of the following:
  - SEGMENT_MARK_CORRUPT (default)
  - SEGMENT_MARK_VALID

```
conn sys as sysdba

declare
 v_segname  varchar2(100) := 'EMPLOYEES';
 v_segowner varchar2(100) :='HR';
 v_tbs  varchar2(100);
 v_fno  number ;
 v_rfno number;
 v_hdr  number;
begin
 -- retreive tablespace name, absolute file number
 select tablespace_name, header_file, header_block
   into v_tbs, v_fno, v_hdr
  from  dba_segments
 where  segment_name=v_segname and owner=v_segowner;

 select relative_fno
   into v_rfno
   from dba_data_files
  where tablespace_name = v_tbs and file_id=v_fno;

 DBMS_SPACE_ADMIN.SEGMENT_CORRUPT (
  tablespace_name      =>v_tbs,
  header_relative_file =>v_rfno,
  header_block         =>v_hdr,
  corrupt_option       =>DBMS_SPACE_ADMIN.SEGMENT_MARK_CORRUPT);
end;
```

Dropping a Corrupted Segment

- Use SEGMENT_DROP_CORRUPT to drop a segment currently marked corrupt (without reclaiming space). The space for the segment is not released, and it must be fixed by using the TABLESPACE_FIX_BITMAPS Procedure or the TABLESPACE_REBUILD_BITMAPS Procedure.
- If the segment state is valid, the procedure returns ORA-03211 error.

```
conn sys as sysdba

declare
 v_segname  varchar2(100) := 'EMP';
 v_segowner varchar2(100) :='HR';
 v_tbs  varchar2(100);
 v_fno  number ;
```

```
 v_rfno number;
 v_hdr  number;
begin
 -- retreive tablespace name, absolute file number
 select tablespace_name, header_file, header_block
   into v_tbs, v_fno, v_hdr
  from  dba_segments
 where  segment_name=v_segname and owner=v_segowner;

 select relative_fno
   into v_rfno
   from dba_data_files
  where tablespace_name = v_tbs and file_id=v_fno;

 DBMS_SPACE_ADMIN.SEGMENT_DROP_CORRUPT(
  tablespace_name =>v_tbs,
  header_relative_file =>v_rfno,
  header_block => v_hdr);
end;
```

Dumping a Segment Header and Bitmap Blocks

- Use DBMS_SPACE_ADMIN.SEGMENT_DUMP
- The dump file named sid_ora_process_id.trc is generated in the location specified in the USER_DUMP_DEST

```
conn sys as sysdba
declare
 v_segname  varchar2(100) := 'NAMES';
 v_segowner varchar2(100) :='HR';
 v_tbs  varchar2(100);
 v_fno  number;
 v_rfno number;
 v_hdr  number;
begin
 -- retreive tablespace name, absolute file number
 select tablespace_name, header_file, header_block
   into v_tbs, v_fno, v_hdr
  from  dba_segments
 where  segment_name=v_segname and owner=v_segowner;

 select relative_fno
   into v_rfno
   from dba_data_files
  where tablespace_name = v_tbs and file_id=v_fno;

 DBMS_SPACE_ADMIN.SEGMENT_DUMP(
  tablespace_name =>v_tbs,
  header_relative_file =>v_rfno,
  header_block =>v_hdr,
  dump_option  => DBMS_SPACE_ADMIN.SEGMENT_DUMP_EXTENT_MAP);
end;
```

Marking a DBA Range in Bitmap as Free or Used

- The procedure TABLESPACE_FIX_BITMAPS marks the appropriate DBA range (extent) as free or used in bitmap.
- The BEGIN and END blocks should be in extent boundary and should be extent multiple.
- fix_option takes one of the following

- TABLESPACE_EXTENT_MAKE_FREE
- TABLESPACE_EXTENT_MAKE_USED

```
conn sys as sysdba
DBMS_SPACE_ADMIN.TABLESPACE_FIX_BITMAPS (
 tablespace_name =>'EXAMPLE',
 dbarange_relative_file =>4,
 dbarange_begin_block =>33,
 dbarange_end_block =>83,
 fix_option =>DBMS_SPACE_ADMIN.TABLESPACE_EXTENT_MAKE_FREE);
```

Rebuilding the Appropriate Bitmap

```
conn sys
begin
DBMS_SPACE_ADMIN.TABLESPACE_REBUILD_BITMAPS (
 tablespace_name =>'EXAMPLE',
 bitmap_relative_file =>NULL, -- all files
 bitmap_block =>NULL); -- Block number of bitmap block to rebuild
                       -- NULL = all blocks
end;
```

Rebuilding Quotas for Given Tablespace

```
conn sys
exec DBMS_SPACE_ADMIN.TABLESPACE_REBUILD_QUOTAS('USERS');
```

Migrating from a Dictionary-Managed to a Locally Managed Tablespace

```
-- This operation is done online, but space management operations are blocked
-- ASSM won't be active on migrated objects
conn sys
EXEC DBMS_SPACE_ADMIN.TABLESPACE_MIGRATE_TO_LOCAL ('USERS');

-- another way requiring table lock (better)
ALTER TABLE emp MOVE TABLESPACE tbsp_new;
ALTER INDEX emp_pk_idx REBUILD TABLESPACE tbsp_idx_new;
```

Fixing the State of the Segments in A Tablespace

Use TABLESPACE_FIX_SEGMENT_STATES to fix the state of the segments in a tablespace in which migration was aborted.

```
conn sys
EXECUTE DBMS_SPACE_ADMIN.TABLESPACE_FIX_SEGMENT_STATES('TS1')
```

Scenario 1: Fixing Bitmap When Allocated Blocks are Marked Free (No Overlap)

The TABLESPACE_VERIFY procedure discovers that a segment has allocated blocks that are marked free in the bitmap, but no overlap between segments is reported.

In this scenario, perform the following tasks:

1. Call the SEGMENT_DUMP procedure to dump the ranges that the administrator allocated to the segment.

2. For each range, call the TABLESPACE_FIX_BITMAPS procedure with the TABLESPACE_EXTENT_MAKE_USED option to mark the space as used.

3. Call TABLESPACE_REBUILD_QUOTAS to fix up quotas.

Scenario 2: Dropping a Corrupted Segment

You cannot drop a segment because the bitmap has segment blocks marked "free". The system has automatically marked the segment corrupted. In this scenario, perform the following tasks:

1. Call the SEGMENT_VERIFY procedure with the SEGMENT_VERIFY_EXTENTS_GLOBAL option. If no overlaps are reported, then proceed with steps 2 through 5.

2. Call the SEGMENT_DUMP procedure to dump the DBA ranges allocated to the segment.

3. For each range, call TABLESPACE_FIX_BITMAPS with the TABLESPACE_EXTENT_MAKE_FREE option to mark the space as free.

4. Call SEGMENT_DROP_CORRUPT to drop the SEG$ entry.

5. Call TABLESPACE_REBUILD_QUOTAS to fix up quotas.

Scenario 3: Fixing Bitmap Where Overlap is Reported

The TABLESPACE_VERIFY procedure reports some overlapping. Some of the real data must be sacrificed based on previous internal errors. After choosing the object to be sacrificed, in this case say, table t1, perform the following tasks:

1. Make a list of all objects that t1 overlaps.

2. Drop table t1. If necessary, follow up by calling the SEGMENT_DROP_CORRUPT procedure.

3. Call the SEGMENT_VERIFY procedure on all objects that t1 overlapped. If necessary, call the TABLESPACE_FIX_BITMAPS procedure to mark appropriate bitmap blocks as used.

4. Rerun the TABLESPACE_VERIFY procedure to verify the problem is resolved.

Scenario 4: Correcting Media Corruption of Bitmap Blocks

A set of bitmap blocks has media corruption. In this scenario, perform the following tasks:

1. Call the TABLESPACE_REBUILD_BITMAPS procedure, either on all bitmap

blocks, or on a single block if only one is corrupt.

2. Call the TABLESPACE_REBUILD_QUOTAS procedure to rebuild quotas.

3. Call the TABLESPACE_VERIFY procedure to verify that the bitmaps are consistent.

Transporting Tablespaces Between Databases

Limitations on Transportable Tablespace Use

- The source and target database must use the same character set and national character set.
- Objects with underlying objects (such as materialized views) or contained objects (such as partitioned tables) are not transportable unless all of the underlying or contained objects are in the tablespace set.
- You cannot transport the SYSTEM tablespace or objects owned by the user SYS. This means that you cannot use TTS for PL/SQL, triggers, or views. These would have to be moved with export.
- You cannot transport a table with a materialized view unless the mview is in the transport set you create.
- You cannot transport a partition of a table without transporting the entire table.

1. Check *endian* format of both platforms.
   For cross-platform transport, check the endian format of both platforms by querying the V$TRANSPORTABLE_PLATFORM view.

   You can find out your own platform name:

```
select platform_name from v$database
```

2. Pick a self-contained set of tablespaces.

   The following statement can be used to determine whether tablespaces sales_1 and sales_2 are self-contained, with referential integrity constraints taken into consideration:

```
DBMS_TTS.TRANSPORT_SET_CHECK( TS_LIST =>'sales_1,sales_2', INCL_CONSTRAINTS
=>TRUE, FULL_CHECK =>TRUE)
```

   Note: You must have been granted the EXECUTE_CATALOG_ROLE role (initially signed to SYS) to execute this procedure.

   You can see all violations by selecting from the TRANSPORT_SET_VIOLATIONS view. If the set of tablespaces is self-contained, this view is empty.

3. Generate a transportable *tablespace set*.

   3.1. Make all tablespaces in the set you are copying read-only.

   3.2. Export the metadata describing the objects in the tablespace(s)

```
EXPDP system/password DUMPFILE=expdat.dmp DIRECTORY=dpump_dir
TRANSPORT_TABLESPACES = sales_1,sales_2
TRANSPORT_FULL_CHECK=Y
```

   3.3. If you want to convert the tablespaces in the source database, use the RMAN

```
RMAN TARGET /
CONVERT TABLESPACE sales_1,sales_2
TO PLATFORM 'Microsoft Windows NT'
FORMAT '/temp/%U'
```

4. Transport the tablespace set.

   Transport both the datafiles and the export file of the tablespaces to a place accessible to the target database.

5. Convert tablespace set, if required, in the destination database.

```
Use RMAN as follows:
RMAN> CONVERT DATAFILE
'/hq/finance/work/tru/tbs_31.f',
'/hq/finance/work/tru/tbs_32.f',
'/hq/finance/work/tru/tbs_41.f'
TO PLATFORM="Solaris[tm] OE (32-bit)"
FROM PLATFORM="HP TRu64 UNIX"
DBFILE_NAME_CONVERT=
"/hq/finance/work/tru/", "/hq/finance/dbs/tru"
PARALLELISM=5
```

   Note: The source and destination platforms are optional.

   Note: By default, Oracle places the converted files in the Flash Recovery Area, without changing the datafile names.

   Note: If you have CLOB data on a small-endian system in an Oracle database version before 10g and with a varying-width character set and you are transporting to a database in a big-endian system, the CLOB data must be converted in the destination database. RMAN does not handle the conversion during the CONVERT phase. However, Oracle database automatically handles the conversion while accessing the CLOB data.

   If you want to eliminate this run-time conversion cost from this automatic conversion, you can issue the CREATE TABLE AS SELECT command before accessing the data.

6. Plug in the tablespace.

```
IMPDP system/password DUMPFILE=expdat.dmp DIRECTORY=dpump_dir
TRANSPORT_DATAFILES=
/salesdb/sales_101.dbf,
/salesdb/sales_201.dbf
REMAP_SCHEMA=(dcranney:smith) REMAP_SCHEMA=(jfee:williams)
```

If required, put the tablespace into `READ WRITE` mode.

Using Transportable Tablespaces: Scenarios

Transporting and Attaching Partitions for Data Warehousing

1. In a staging database, you create a new tablespace and make it contain the table you want to transport. It should have the same columns as the destination partitioned table.
2. Create an index on the same columns as the local index in the partitioned table.
3. Transport the tablespace to the data warehouse.
4. In the data warehouse, add a partition to the table.
   ```
   ALTER TABLE sales ADD PARTITION jul98 VALUES LESS THAN (1998, 8, 1)
   ```
5. Attach the transported table to the partitioned table by exchanging it with the new partition:
   ```
   ALTER TABLE sales EXCHANGE PARTITION jul98 WITH TABLE jul_sales
   INCLUDING INDEXES WITHOUT VALIDATION
   ```

Publishing Structured Data on CDs

A data provider can load a tablespace with data to be published, generate the transportable set, and copy the transportable set to a CD. When customers receive this CD, they can plug it into an existing database without having to copy the datafiles from the CD to disk storage.

Note: In this case, it is highly recommended to set the `READ_ONLY_OPEN_DELAYED` initialization parameter to `TRUE`.

Moving Databases Across Platforms Using Transportable Tablespaces

You can use the transportable tablespace feature to migrate a database to a different platform.

However, you cannot transport the SYSTEM tablespace. Therefore, objects such as sequences, PL/SQL packages, and other objects that depend on the SYSTEM tablespace are not transported. You must either create these objects manually on the destination database, or use Data Pump to transport the objects that are not moved by transportable tablespace.

Managing Alert Thresholds

Getting the Current Threshold Setting

- List of supported metrics can be found in the documentation "PL/SQL Packages and Types Reference": 10g, 11g or link or from the query below:

```
SELECT METRIC_ID, METRIC_NAME, METRIC_UNIT,
GROUP_ID, GROUP_NAME
FROM V$METRICNAME
ORDER BY METRIC_NAME
```

```
-- current threshold settings
select * from DBA_THRESHOLDS;
```

```
set serveroutput on
DECLARE
 V_WOPERATOR   BINARY_INTEGER;
 V_WVALUE      VARCHAR2(50);
```

```
 V_COPERATOR   BINARY_INTEGER;
 V_CVALUE      VARCHAR2(50);
 V_OBS_PERIOD BINARY_INTEGER;
 V_CON_PERIOD BINARY_INTEGER;
 FUNCTION GET_OPERATOR_NAME( P_OPER IN BINARY_INTEGER) RETURN VARCHAR2
 IS
 BEGIN
  IF P_OPER =0 THEN
   RETURN 'GT';
  ELSIF P_OPER =1 THEN
   RETURN 'EQ';
  ELSIF P_OPER =2 THEN
   RETURN 'LT';
  ELSIF P_OPER =3 THEN
   RETURN 'LE';
  ELSIF P_OPER =4 THEN
   RETURN 'GE';
  ELSIF P_OPER =5 THEN
   RETURN 'OPERATOR_CONTAINS';
  ELSIF P_OPER =6 THEN
   RETURN 'NE';
  ELSIF P_OPER =7 THEN
   RETURN 'OPERATOR_DO_NOT_CHECK';
  END IF;
 END GET_OPERATOR_NAME;
BEGIN
DBMS_SERVER_ALERT.GET_THRESHOLD(
 metrics_id =>DBMS_SERVER_ALERT.TABLESPACE_PCT_FULL,
 warning_operator      =>V_WOPERATOR ,
 warning_value         =>V_WVALUE,
 critical_operator =>V_COPERATOR,
 critical_value        =>V_CVALUE,
 observation_period =>V_OBS_PERIOD,
 consecutive_occurrences =>V_CON_PERIOD,
 instance_name =>NULL,
 object_type           => DBMS_SERVER_ALERT.OBJECT_TYPE_TABLESPACE,
 object_name   =>NULL);
 DBMS_OUTPUT.PUT_LINE('Warning  Op.: '|| GET_OPERATOR_NAME(V_WOPERATOR));
 DBMS_OUTPUT.PUT_LINE('Warning  Val: '||V_WVALUE);
 DBMS_OUTPUT.PUT_LINE('Critical Op.: '|| GET_OPERATOR_NAME(V_COPERATOR));
 DBMS_OUTPUT.PUT_LINE('Critical Val: '||V_CVALUE);
EXCEPTION
 WHEN OTHERS THEN
 IF SQLCODE='-13799' THEN
  DBMS_OUTPUT.PUT_LINE('No threshold was found with the specified threshold
key.(ORA-13799)');
 ELSE
  RAISE;
 END IF;
END;
/
```

Setting Tablespace Alert Thresholds

- warning_operator takes one of the following (not all applicable for all metrics):
  - OPERATOR_EQ GE GT LE LT NE
  - OPERATOR_CONTAINS

- OPERATOR_DO_NOT_CHECK (disables the alert for the specified metric)

```
-- set the free-space-remaining thresholds in the USERS tablespace to 10 MB
(warning)
--  and 2 MB (critical), and disable the percent-full thresholds.
BEGIN
DBMS_SERVER_ALERT.SET_THRESHOLD(
 metrics_id => DBMS_SERVER_ALERT.TABLESPACE_BYT_FREE, -- Tablespace FREE
space in KB
 warning_operator => DBMS_SERVER_ALERT.OPERATOR_LT, -- GT is not applicable
here
 warning_value => '10240',
 critical_operator => DBMS_SERVER_ALERT.OPERATOR_LT,
 critical_value => '2048',
 observation_period => 1, -- computation period (1-60 min)
 consecutive_occurrences => 1, -- violation times before alert
 instance_name => NULL, -- NULL= 'database_wide'. Passed value is not checked
 object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_TABLESPACE,
 object_name => 'USERS'); -- if NULL -> All Tablespaces

DBMS_SERVER_ALERT.SET_THRESHOLD(
 metrics_id => DBMS_SERVER_ALERT.TABLESPACE_PCT_FULL, -- tablespace USAGE by
%
 warning_operator => DBMS_SERVER_ALERT.OPERATOR_DO_NOT_CHECK,
 warning_value => '0',
 critical_operator => DBMS_SERVER_ALERT.OPERATOR_DO_NOT_CHECK,
 critical_value => '0',
 observation_period => 1,
 consecutive_occurrences => 1,
 instance_name => NULL,
 object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_TABLESPACE,
 object_name => 'USERS');
END;
/

SELECT *
FROM dba_thresholds
where object_name = 'USERS' and object_type='TABLESPACE';
```

Restoring a Tablespace to Database Default Thresholds

- You can restore the metric threshold values to revert to the database defaults by setting them to NULL in the DBMS_SERVER_ALERT.SET_THRESHOLD.

```
BEGIN
DBMS_SERVER_ALERT.SET_THRESHOLD(
 metrics_id => DBMS_SERVER_ALERT.TABLESPACE_PCT_FULL, -- tablespace USAGE by
%
 warning_operator => NULL, -- do not use ''
 warning_value => NULL,
 critical_operator => NULL,
 critical_value => NULL,
 observation_period => 1,
 consecutive_occurrences => 1,
 instance_name => NULL,
 object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_TABLESPACE,
 object_name => 'USERS');
END;
/
```

```
SELECT *
FROM dba_thresholds
where object_name = 'USERS' and object_type='TABLESPACE';
```

Modifying Database Default Thresholds

- set the object_name to NULL

```
BEGIN
DBMS_SERVER_ALERT.SET_THRESHOLD(
 metrics_id => DBMS_SERVER_ALERT.TABLESPACE_PCT_FULL, -- redo with
TABLESPACE_BYT_FREE
 warning_operator => DBMS_SERVER_ALERT.OPERATOR_GT,
 warning_value => '80',
 critical_operator => DBMS_SERVER_ALERT.OPERATOR_GT,
 critical_value => '92',
 observation_period => 1,
 consecutive_occurrences => 1,
 instance_name => NULL,
 object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_TABLESPACE,
 object_name => NULL);
END;
/

SELECT *
FROM dba_thresholds
where object_name = 'USERS' and object_type='TABLESPACE';
```

Viewing Alerts

```
-- outstanding alerts (to be cleared)
SELECT
 SEQUENCE_ID,OWNER,
 OBJECT_NAME,OBJECT_TYPE,SUBOBJECT_NAME,
 REASON_ID, REASON,TIME_SUGGESTED,
 SUGGESTED_ACTION,ADVISOR_NAME,METRIC_VALUE,
 MESSAGE_TYPE,MESSAGE_GROUP,MESSAGE_LEVEL,
 HOSTING_CLIENT_ID,MODULE_ID,PROCESS_ID,
 HOST_ID,HOST_NW_ADDR,INSTANCE_NAME,
 INSTANCE_NUMBER,USER_ID,EXECUTION_CONTEXT_ID,CREATION_TIME
FROM DBA_OUTSTANDING_ALERTS

-- history of alerts that have been cleared
SELECT
 SEQUENCE_ID,OWNER,
 OBJECT_NAME,SUBOBJECT_NAME,OBJECT_TYPE,
 REASON_ID, REASON,TIME_SUGGESTED,
 SUGGESTED_ACTION,ADVISOR_NAME,METRIC_VALUE,
 MESSAGE_TYPE,MESSAGE_GROUP,MESSAGE_LEVEL,
 HOSTING_CLIENT_ID,MODULE_ID,PROCESS_ID,
 HOST_ID,HOST_NW_ADDR,INSTANCE_NAME,INSTANCE_NUMBER,
 USER_ID,EXECUTION_CONTEXT_ID,CREATION_TIME
FROM DBA_ALERT_HISTORY
ORDER BY SEQUENCE_ID

-- list of all metrics
SELECT METRIC_ID, METRIC_NAME, METRIC_UNIT,GROUP_ID, GROUP_NAME
FROM V$METRICNAME
```

```
 ORDER BY METRIC_NAME

 -- system-level metric values in memory
 SELECT BEGIN_TIME,END_TIME,INTSIZE_CSEC,
  GROUP_ID,ENTITY_ID,ENTITY_SEQUENCE,
  METRIC_ID,METRIC_NAME,VALUE,METRIC_UNIT
 FROM V$METRIC -- also V$METRIC_HISTORY
 ORDER BY BEGIN_TIME, VALUE DESC

 -- alert types
 select
 INST_ID,REASON_ID,OBJECT_TYPE,TYPE,GROUP_NAME,SCOPE,INTERNAL_METRIC_CATEGORY,
 INTERNAL_METRIC_NAME
 from GV$ALERT_TYPES
 order by OBJECT_TYPE,TYPE
```

Managing Datafiles and Tempfiles

Creating Datafiles

```
CREATE TABLESPACE
CREATE TEMPORARY TABLESPACE ge 8-9
ALTER TABLESPACE ... ADD DATAFILE
ALTER TABLESPACE ... ADD TEMPFILE ge 8-9
CREATE DATABASE
ALTER DATABASE ... CREATE DATAFILE
```

Enabling and Disabling Automatic Extension for a Datafile

```
ALTER TABLESPACE users
ADD DATAFILE '/u02/oracle/rbdb1/users03.dbf' SIZE 10M
AUTOEXTEND ON
NEXT 512K
MAXSIZE 250M;
```

Manually Resizing a Datafile

```
-- if there is space in the datafile
ALTER DATABASE DATAFILE '/u02/oracle/rbdb1/stuff01.dbf' RESIZE 100M;
Bringing Datafiles Online or Taking Offline in ARCHIVELOG Mode
ALTER DATABASE DATAFILE '/u02/oracle/rbdb1/stuff01.dbf' ONLINE;
ALTER DATABASE DATAFILE '/u02/oracle/rbdb1/stuff01.dbf' OFFLINE;
```

Bringing Datafiles Online or Taking Offline in ARCHIVELOG Mode

```
ALTER DATABASE DATAFILE ... {ONLINE|OFFLINE}
-- all datafiles will be affected in the following code
--  this is different from ALTER TABLESPACE...ONLINE|OFFLINE which controls
tablespace availability
ALTER TABLESPACE ... DATAFILE {ONLINE|OFFLINE}
ALTER TABLESPACE ... TEMPFILE {ONLINE|OFFLINE}
```

Taking Datafiles Offline in NOARCHIVELOG Mode

- Use it when you want to drop the datafile.

```
-- datafile cannot be brought ONLINE again
ALTER DATABASE DATAFILE ... OFFLINE FOR DROP;
-- the datafile MUST then be dropped
--  for dictionary managed tablespace
ALTER TABLESPACE ... DROP DATAFILE
DROP TABLESPACE ... INCLUDING CONTENTS AND DATAFILES
```

Renaming and Relocating Datafiles in a Single Tablespace

```
ALTER TABLESPACE users OFFLINE NORMAL;
Copy the datafiles to their new locations and rename them using the operating
system.
ALTER TABLESPACE users RENAME DATAFILE
  '/u02/oracle/rbdb1/user1.dbf',
  '/u02/oracle/rbdb1/user2.dbf'
 TO
  '/u02/oracle/rbdb1/users01.dbf',
  '/u02/oracle/rbdb1/users02.dbf';
-- for system, default temporary, or undo
--  ALTER TABLESPACE cannot be used because you cannot take them OFFLINE
mount the database
ALTER DATABASE RENAME FILE '/u02/oracle/rbdb1/sort01.dbf',
 '/u02/oracle/rbdb1/user3.dbf'
TO '/u02/oracle/rbdb1/temp01.dbf', '/u02/oracle/rbdb1/users03.dbf;
Back up the database.
```

Dropping Datafiles

- The following restrictions apply:
  - The database must be open.
  - The datafile must be empty, otherwise use drop the tablespace.
  - You cannot drop datafiles in a read-only tablespace.
  - You cannot drop datafiles in the SYSTEM tablespace.
  - If a datafile in a locally managed tablespace is offline, it cannot be dropped.

```
ALTER TABLESPACE example DROP DATAFILE ...
ALTER TABLESPACE lmtemp DROP TEMPFILE
ALTER DATABASE TEMPFILE .. DROP INCLUDING DATAFILES
```

Copying a File on a Local File System

- The copied file must meet the following requirements:
  - The size must be a multiple of 512 bytes.
  - The size must be less than or equal to two terabytes.
- Be aware not to coy a file that is being used by a process.
- If you are copying a database datafile, make it READ ONLY before you start to copy.

```
CREATE DIRECTORY SOURCE_DIR AS '/usr/admin/source';
CREATE DIRECTORY DEST_DIR AS '/usr/admin/destination';
GRANT READ ON DIRECTORY source_dir TO strmadmin;
GRANT WRITE ON DIRECTORY dest_dir TO strmadmin;
CONNECT strmadmin/strmadminpw
BEGIN
DBMS_FILE_TRANSFER.COPY_FILE(
 source_directory_object => 'SOURCE_DIR',
 source_file_name => 'db1.dat',
```

```
 destination_directory_object => 'DEST_DIR',
 destination_file_name => 'db1_copy.dat');
END;
```

Transferring a File to a Different Database

- In order to transfer a file the other way around, you must replace the **PUT_FILE** procedure with the GET_FILE procedure.
- If you are copying a database datafile, make it READ ONLY before you start to copy.
- You can monitor copying progress using V$SESSION_LONGOPS view.

```
CREATE DATABASE LINK ODB
 CONNECT TO system IDENTIFIED BY system_passwd USING 'prod1';

BEGIN
DBMS_FILE_TRANSFER.PUT_FILE(
 SOURCE_DIRECTORY_OBJECT => 'SOURCE_DIR',
 SOURCE_FILE_NAME => 'mydata1.dbf',
 DESTINATION_DIRECTORY_OBJECT => 'DEST_DIR',
 DESTINATION_FILE_NAME => 'mydata2.dbf'
 DESTINATION_DATABASE => 'ODB.ACME.COM');
END;
/
BEGIN
DBMS_FILE_TRANSFER.GET_FILE(
 SOURCE_DIRECTORY_OBJECT => 'SOURCE_DIR',
 SOURCE_FILE_NAME => 'TEST01.DBF',
 SOURCE_DATABASE => 'ODB.ACME.COM',
 DESTINATION_DIRECTORY_OBJECT => 'DEST_DIR',
 DESTINATION_FILE_NAME => 'TEST01.DBF');
END;
/
```

Dumping a Data Block

```
ALTER SYSTEM DUMP DATAFILE 3 BLOCK 3281;

-- To dump a number of consecutive blocks::
ALTER SYSTEM DUMP DATAFILE 5 BLOCK MIN 42 BLOCK MAX 50;

select value from v$parameter where name='user_dump_dest';
/u01/app/oracle/admin/pasu/udump/pasu_ora_29673.trc



/* to dump index blocks */
-- get object id of the index:
SELECT object_id FROM dba_objects WHERE object_name = 'MYINDEX';

-- do a treedump of the index:
ALTER SESSION SET EVENTS 'immediate trace name treedump level 106315';

                                          index height
        distinct index blocks in the lower level |
        RBA  block at position zero   |          |
        |                   |         |          |
 branch: 0×1c3588a 29579402 (0: nrow: 222, level: 1)


                           number of entries
 lead block number (starts from -1)     |     non-deleted entries
                              |         |         |
```

```
   leaf: 0×1c3588b 29579403 (-1: nrow: 485 rrow: 485)
   leaf: 0×1c3588c 29579404 (0: nrow: 479 rrow: 479)
   leaf: 0×1c3588d 29579405 (1: nrow: 479 rrow: 479)
   leaf: 0×1c3588e 29579406 (2: nrow: 479 rrow: 479)


-- define the RBA of the block to dump then
--  get its file# and block#:
SELECT DBMS_UTILITY.DATA_BLOCK_ADDRESS_FILE(223456765),
 DBMS_UTILITY.DATA_BLOCK_ADDRESS_BLOCK(223456765)
 FROM dual;


-- dump the header block
ALTER SYSTEM DUMP DATAFILE 7 BLOCK 328745;


-- also, you can find the root block ( it is the block following the header bloc):
SELECT header_file, header_block+1 FROM dba_segments WHERE segment_name='MYINDEX';
```

Managing Undo Tablespaces

Obtaining Information on Undo

```
 SHOW PARAMETER UNDO
 NAME TYPE VALUE
 ----------------- ------- -------------
 undo_management string AUTO
 undo_retention integer 900
 undo_tablespace string UNDOTBS01

 -- undo usage in 10-min-intervals for last 7 days
 select TO_CHAR(BEGIN_TIME, 'MM/DD/YYYY HH24:MI:SS') BEGIN_TIME,
  TO_CHAR(END_TIME, 'MM/DD/YYYY HH24:MI:SS') END_TIME,
  UNDOBLKS UNDO_BLOCKS,
  MAXQUERYLEN MAX_QUERY_LENGTH_INSEC,
  MAXQUERYID LONGEST_QUERY_ID,
  TXNCOUNT TOTAL_TRANSACTIONS,
  MAXCONCURRENCY
  from V$UNDOSTAT
 order by BEGIN_TIME DESC

 -- stats history of V$UNDOSTAT
 select
  TO_CHAR(BEGIN_TIME, 'MM/DD/YYYY HH24:MI:SS') BEGIN_TIME,
  TO_CHAR(END_TIME, 'MM/DD/YYYY HH24:MI:SS') END_TIME,
  INSTANCE_NUMBER,SNAP_ID,
  UNDOBLKS,
  TXNCOUNT TOTAL_TRANSACTIONS,
  MAXQUERYLEN MAX_QUERY_LENGTH_INSEC,
  MAXQUERYSQLID,
  MAXCONCURRENCY,
  SSOLDERRCNT ORA01555_CNT,
  NOSPACEERRCNT NOSPACE_CNT,
  ACTIVEBLKS,UNEXPIREDBLKS
 from DBA_HIST_UNDOSTAT


 -- undo segments
 select SEGMENT_NAME,OWNER,STATUS from DBA_ROLLBACK_SEGS;
```

```
-- undo extents
select
 e.SEGMENT_NAME,
 e.TABLESPACE_NAME,
 e.EXTENT_ID,
 e.FILE_ID,
 e.BLOCK_ID,
 e.BYTES/1024 SIZE_KB,
 e.RELATIVE_FNO,
 e.STATUS -- ACTIVE EXPIRED UNEXPIRED
from DBA_UNDO_EXTENTS e
order by e.STATUS ASC

-- undo sizes by STATUS
select
 e.TABLESPACE_NAME,
 e.STATUS,
 to_char(sum(e.BYTES/1024),'999,999,999,999') SIZE_KB
from DBA_UNDO_EXTENTS e
group by e.TABLESPACE_NAME, e.STATUS
order by e.STATUS


-- undo sizes consumed by active transactions by username
SELECT s.username, sum(t.used_ublk) used_undo_blocks
from   v$session s, v$transaction t
where  s.saddr = t.ses_addr
 and   t.status='ACTIVE'
group by s.username
order by s.username
```

Enabling Automatic Undo Management

```
select value from v$parameter where upper(name)='UNDO_MANAGEMENT';
alter system set UNDO_MANAGEMENT=AUTO scope=spfile ;
```

Creating an Undo Tablespace

```
CREATE UNDO TABLESPACE undotbs2
DATAFILE '/u01/oracle/rbdb1/undo0201.dbf' SIZE 200M REUSE AUTOEXTEND ON;
```

Setting Startup Undo Tablespace

```
select value from v$parameter where upper(name)='UNDO_TABLESPACE';
-- if there are multiple undo tablespaces
--  after all active transactions have committed, the undo tablespace automatically
--  goes from the PENDING OFFLINE mode to the OFFLINE mode.
alter system set UNDO_TABLESPACE='UNDOTBS1';
-- the following switches out current undo tablespace to the next available one
alter system set UNDO_TABLESPACE = '';
```

Tuning Undo Retention

- If the undo tablespace is configured with the AUTOEXTEND option, undo retention tuning is slightly different. In this case, the database tunes the undo retention period to be slightly longer than the longest-running query on the system at that time.
- For a fixed size undo tablespace, the database tunes the undo retention period based on 85% of the tablespace size, or on the warning alert threshold percentage for space used, whichever is lower.
- If you want a fixed undo size, use the Undo Advisor to specify the proper undo size for your requirement. You should estimate:

- The length of your expected longest running query
- The longest interval that you will require for flashback operations

```
-- determine the current retention period
select to_char(begin_time, 'DD-MON-RR HH24:MI') begin_time,
to_char(end_time, 'DD-MON-RR HH24:MI') end_time,
tuned_undoretention
from v$undostat order by end_time;

-- maximum query time
SELECT round(MAX(maxquerylen)/60) Minutes FROM v$undostat;
```

Using Undo Advisor

- Use OEM or PL/SQL

```
-- retreive available AWR snaps
select SNAP_ID, STARTUP_TIME, BEGIN_INTERVAL_TIME, END_INTERVAL_TIME
from DBA_HIST_SNAPSHOT
order by SNAP_ID;

set serveroutput on
DECLARE
 tid NUMBER; -- task ID
 tname VARCHAR2(30); -- task name
 oid NUMBER;
BEGIN
 DBMS_ADVISOR.CREATE_TASK('Undo Advisor', tid, tname, 'Undo Advisor Task');
 DBMS_ADVISOR.CREATE_OBJECT(tname, 'UNDO_TBS', null, null, null, null, oid);
 DBMS_ADVISOR.SET_TASK_PARAMETER(tname, 'TARGET_OBJECTS', oid);
 DBMS_ADVISOR.SET_TASK_PARAMETER(tname, 'START_SNAPSHOT', 52);
 DBMS_ADVISOR.SET_TASK_PARAMETER(tname, 'END_SNAPSHOT', 56);
 DBMS_ADVISOR.SET_TASK_PARAMETER(tname, 'INSTANCE', 1);
 DBMS_ADVISOR.execute_task(tname);
 DBMS_OUTPUT.PUT_LINE(tname);
end;
/

SELECT DBA_ADVISOR_RECOMMENDATIONS.GET_TASK_REPORT('TASKNAME')
FROM DUAL;
```

Setting the Undo Retention Period

- You must set UNDO_RETENTION parameter when:
  - The undo tablespace has the AUTOEXTEND option enabled
  - You want to set undo retention for LOBs
  - You want retention guarantee

```
select value from v$parameter where upper(name)='UNDO_RETENTION';
alter system set UNDO_RETENTION = 2400;
```

Enabling Retention Guarantee

- Enabling retention guarantee can cause multiple DML operations to fail. Use with caution.

```
select RETENTION from DBA_TABLESPACES where TABLESPACE_NAME='UNDOTBS1';
create undo tablespace undotbs01 .. RETENTION GUARANTEE;
alter tablespace undotbs1 RETENTION GUARANTEE;
alter tablespace undotbs1 RETENTION NOGUARANTEE;
```

Dropping an Undo Tablespace

```
DROP TABLESPACE undotbs_01;
```

To Drop a Corrupt UNDO Tablespace

An Undo tablespace containing a corrupted undo rollback segment may lead to one or more of the following:

- ORA-00376: file xx cannot be read at this time
- SIMON process being hang waiting for the event "wait for a undo record" forever!

```
Identify the bad rollback segment(s)(Oracle would report it anyway):
-- looking for NEEDS RECOVERY or PARTIALLY AVAILABLE
select
segment_name, status
from dba_rollback_segs
where tablespace_name='undotbs_corrupt'
and status not in ('OFFLINE', 'ONLINE');


-- Let's say it's _SYSSMU9$


-- (optionally) create a new undo tablespace
--   and replace it with the current one
CREATE UNDO TABLESPACE undotbs2
DATAFILE '/u01/oracle/rbdb1/undo0201.dbf' SIZE 200M REUSE AUTOEXTEND ON;
alter system set UNDO_TABLESPACE='UNDOTBS2';

-- Next, create pfile and in it:
undo_management = MANUAL
_offline_rollback_segments=_SYSSMU9$


-- shutdown the db

-- open it using the pfile

-- drop the offending rollback segment
--  (and the undo tablespace altogether if the other one was created)
-- Note: datafiles might not actually be deleted, check them
drop tablespace .. including contents and datafiles


-- bounce the db (make it start using the spfile)
--  this means it will go back to using the automatic undo
```

Using Oracle Managed Files (OMF)

- OMF datafiles have to be created in one directory

```
set DB_CREATE_FILE_DEST, DB_CREATE_ONLINE_LOG_DEST_n, and DB_RECOVERY_FILE_DEST

-- when OMF enabled
CREATE TABLESPACE finance01;
ALTER TABLESPACE finance01 ADD DATAFILE 500M;
DROP TABLESPACE finance01; -- dbf auto deleted

-- create db with OMF
db_name=mydb
DB_CREATE_FILE_DEST = '/u01/app/oracle/oradata'
DB_RECOVERY_FILE_DEST_SIZE = 100M
DB_RECOVERY_FILE_DEST = '/u04/app/oracle/oradata'
LOG_ARCHIVE_DEST_1 = 'LOCATION = USE_DB_RECOVERY_FILE_DEST'
SQL> connect sys/sys_passwd as sysdba
Connected to an idle instance.
```

```
SQL> STARTUP NOMOUNT PFILE='initmydb.ora';
SQL> CREATE DATABASE mydb;
```

Managing Schema Objects

Chaching Small Tables in Memory

```
 ALTER TABLE hr.countries CACHE;
```

Creating Virtual Columns

```
 CREATE TABLE EMPLOYEES
 ( empno number PRIMARY KEY,
   sal   NUMBER (8,2) NOT NULL,
   annual_sal AS (sal*12),
   CONSTRAINT MaxAnSal CHECK (annual_sal BETWEEN 0 AND 2000000) );
 SELECT TABLE_NAME, COLUMN_NAME, DATA_DEFAULT
 FROM DBA_TAB_COLUMNS
 WHERE TABLE_NAME='EMPLOYEES' AND COLUMN_NAME='ANNUAL_SAL';

 ALTER TABLE employees ADD (income AS (salary*commission_pct));
```

Creating Partitioned Tables

```
/* range partitions */
CREATE TABLE student_history
(student_id NUMBER(10),
 degree VARCHAR2(3),
 graduation_date DATE,
 final_gpa NUMBER)
 PARTITION BY RANGE (graduation_date)
 (PARTITION p_1997 VALUES LESS THAN
   (TO_DATE('01-JUN-1997','DD-MON-YYYY')) TABLESPACE ts1,
  PARTITION p_1998 VALUES LESS THAN
   (TO_DATE('01-JUN-1998','DD-MON-YYYY')) TABLESPACE ts2,
  PARTITION p_1999 VALUES LESS THAN
   (TO_DATE('01-JUN-1999','DD-MON-YYYY')) TABLESPACE ts3,
  PARTITION p_other VALUES LESS THAN (maxvalue)  TABLESPACE ts4);

CREATE TABLE sales_data
 (ticket_no NUMBER,
 sale_year INT NOT NULL,
 sale_month INT NOT NULL,
 sale_day INT NOT NULL)
 PARTITION BY RANGE (sale_year, sale_month, sale_day)
 (PARTITION sales_q1 VALUES LESS THAN (2008, 04, 01) TABLESPACE ts1,
  PARTITION sales_q2 VALUES LESS THAN (2008, 07, 01) TABLESPACE ts2,
  PARTITION sales_q3 VALUES LESS THAN (2008, 10, 01) TABLESPACE ts3,
  PARTITION sales_q4 VALUES LESS THAN (2009, 01, 01) TABLESPACE ts4);

----------------------------------------------------------------

/*  Interval Partitioning  */
-- You can't use a partitioning key that includes more than one column
--  system generated partions have names SYS_Pn
CREATE TABLE interval_sales
( prod_id NUMBER(6),
  cust_id NUMBER,
  time_id DATE,
..)
```

```
PARTITION BY RANGE (time_id)
INTERVAL(NUMTOYMINTERVAL(1, 'MONTH'))
STORE IN (ts5, ts6, ts7) -- optional
( PARTITION ts1 VALUES LESS THAN (TO_DATE('1-1-2006', 'DD-MM-YYYY')),
  PARTITION ts2 VALUES LESS THAN (TO_DATE('1-1-2007', 'DD-MM-YYYY')),
  PARTITION ts3 VALUES LESS THAN (TO_DATE('1-7-2008', 'DD-MM-YYYY')),
  PARTITION ts4 VALUES LESS THAN (TO_DATE('1-1-2009', 'DD-MM-YYYY')) );

SELECT TABLE_NAME, PARTITION_NAME, PARTITION_POSITION, HIGH_VALUE
FROM Remote DBA_TAB_PARTITIONS
WHERE TABLE_NAME='POS_DATA'
ORDER BY PARTITION_NAME;

create table res (
   res_id      number not null,
   res_date   date,
   hotel_id   number(3),
   guest_id   number )
 partition by range (res_id)
 interval (100) store in (users)
 ( partition p1 values less than (101) );


-- selecting from the generated partition
-- classic method
select * from interval_sales partition for (SYS_P81);
-- the other method (expanded partition access syntax)
select * from
interval_sales partition for (to_date('15-AUG-2009','dd-mon-yyyy'));
alter table res truncate partition for (901);


-- a range partion can be converted to interval partition
alter table pos_data_range set INTERVAL(NUMTOYMINTERVAL(1, 'MONTH'));

-- convert an interval partition to range partition
alter table pos_data_range set INTERVAL();

-- interval can be modified
alter table pos_data set INTERVAL(NUMTOYMINTERVAL(3, 'MONTH'));

-- round robin tablespaces can be modified
alter table pos_data set STORE IN(tablespace1, tablespace2, tablespace3);

----------------------------------------------------------------

/* Hash Partitioning */
-- used when range distribution is not predictable and for high cardinality columns
-- Updates that would cause a record to move across partition boundaries are not allowed
CREATE TABLE sales_data
 (ticket_no NUMBER,
  sale_year INT NOT NULL,
  sale_month INT NOT NULL,
  sale_day INT NOT NULL )
  PARTITION BY HASH (ticket_no)
  PARTITIONS 4
  STORE IN (ts1,ts2,ts3,ts4);

----------------------------------------------------------------

/* List Partitioning */
CREATE TABLE sales_data
 (ticket_no NUMBER,
  sale_year INT NOT NULL,
```

```
  sale_month INT NOT NULL,
  sale_day INT NOT NULL,
  destination_city CHAR(3),
  start_city CHAR(3))
 PARTITION BY LIST (start_city)
 (PARTITION northeast_sales values ('NYC','BOS','PEN') TABLESPACE ts1,
  PARTITION southwest_sales values ('DFW','ORL','HOU') TABLESPACE ts2,
  PARTITION pacificwest_sales values('SAN','LOS','WAS') TABLESPACE ts3,
  PARTITION southeast_sales values ('MIA','CHA','ATL') TABLESPACE ts4);

-----------------------------------------------------------------

/* Reference Partitioning  */
-- put child table data into parent table partitions
-- You can use all partitioning strategies with reference partitioning, except interval
partitioning

CREATE TABLE orders
( order_id NUMBER(12),
 order_date DATE,
 order_mode VARCHAR2(8),
 customer_id NUMBER(6),
 order_status NUMBER(2),
 order_total NUMBER(8,2),
 sales_rep_id NUMBER(6),
 promotion_id NUMBER(6),
 CONSTRAINT orders_pk PRIMARY KEY(order_id))
 PARTITION BY RANGE(order_date)
 ( PARTITION Q1_2005 VALUES LESS THAN (TO_DATE('01-APR-2005','DD-MON-YYYY')),
  PARTITION Q2_2005 VALUES LESS THAN (TO_DATE('01-JUL-2005','DD-MON-YYYY')),
  PARTITION Q3_2005 VALUES LESS THAN (TO_DATE('01-OCT-2005','DD-MON-YYYY')),
  PARTITION Q4_2005 VALUES LESS THAN (TO_DATE('01-JAN-2006','DD-MON-YYYY'))
  );

CREATE TABLE order_items
( order_id NUMBER(12) NOT NULL,
  line_item_id NUMBER(3) NOT NULL,
  product_id NUMBER(6) NOT NULL,
  unit_price NUMBER(8,2),
  quantity NUMBER(8),
  CONSTRAINT order_items_fk
  FOREIGN KEY(order_id) REFERENCES orders(order_id)
 )
PARTITION BY REFERENCE(order_items_fk);

-----------------------------------------------------------------

/* Virtual Column-Based Partitioning */
-- ENABLE ROW MOVEMENT clause ensures row migration among partitions when virtual column
value changes
CREATE TABLE sales
( prod_id NUMBER(6) NOT NULL,
  cust_id NUMBER NOT NULL,
  time_id DATE NOT NULL,
  channel_id CHAR(1) NOT NULL,
  promo_id NUMBER(6) NOT NULL,
  quantity_sold NUMBER(3) NOT NULL,
  amount_sold NUMBER(10,2) NOT NULL,
  total_amount AS (quantity_sold * amount_sold)
)
PARTITION BY RANGE (time_id) INTERVAL (NUMTOYMINTERVAL(1,'MONTH'))
SUBPARTITION BY RANGE(total_amount)
SUBPARTITION TEMPLATE
 ( SUBPARTITION p_small VALUES LESS THAN (1000),
```

```
   SUBPARTITION p_medium VALUES LESS THAN (5000),
   SUBPARTITION p_large VALUES LESS THAN (10000),
   SUBPARTITION p_extreme VALUES LESS THAN (MAXVALUE)
 )
(PARTITION sales_before_2007 VALUES LESS THAN (TO_DATE('01-JAN-2007','dd-MON-yyyy')))
ENABLE ROW MOVEMENT
PARALLEL;

-----------------------------------------------------------------

/* Virtual Column-Based Partitioning */
-- ENABLE ROW MOVEMENT clause ensures row migration among partitions when virtual column
value changes
CREATE TABLE sales
( prod_id NUMBER(6) NOT NULL,
  cust_id NUMBER NOT NULL,
  time_id DATE NOT NULL,
  channel_id CHAR(1) NOT NULL,
  promo_id NUMBER(6) NOT NULL,
  quantity_sold NUMBER(4) NOT NULL,
  amount_sold NUMBER(4) NOT NULL,
  total_amounts AS (quantity_sold * amount_sold)
)
PARTITION BY RANGE (time_id) INTERVAL (NUMTOYMINTERVAL(1,'MONTH'))
SUBPARTITION BY RANGE(total_amounts)
SUBPARTITION TEMPLATE
 ( SUBPARTITION p_small VALUES LESS THAN (1000),
   SUBPARTITION p_medium VALUES LESS THAN (5000),
   SUBPARTITION p_large VALUES LESS THAN (10000),
   SUBPARTITION p_extreme VALUES LESS THAN (MAXVALUE)
 )
(PARTITION sales_before_2007 VALUES LESS THAN (TO_DATE('01-JAN-2007','dd-MON-yyyy')))
ENABLE ROW MOVEMENT
PARALLEL;

-----------------------------------------------------------------

/* System Partitioning */
-- the application decided where to store the data
CREATE TABLE docs
( ID      NUMBER,
  Name    VARCHAR2(255),
  Desc    VARCHAR2(1000))
  PARTITION BY SYSTEM
 ( PARTITION docs_p1 TABLESPACE ts1,
   PARTITION docs_p2 TABLESPACE ts2,
   PARTITION docs_p3 TABLESPACE ts3,
   PARTITION docs_p4 TABLESPACE ts4  );

-- PARTITION must be stated
INSERT INTO docs PARTITION (ts1)
VALUES (1, 'Oracle 11g New Features', 'New features in Oracle 11g Database.');
-- with DELETE command, PARTITION can be stated
DELETE FROM docs PARTITION (ts2) WHERE doc_id=1002;
DELETE FROM docs PARTITION (ts2);
-- PARTITION can be used in queries to target specific partitions
SELECT COUNT(*) FROM  docs PARTITION (ts1)

-----------------------------------------------------------------

/* Range-Hash Partitioning */

CREATE TABLE scout_gear
(equipno NUMBER,equipname VARCHAR(32),price NUMBER)
```

```
 PARTITION BY RANGE (equipno)
 SUBPARTITION BY HASH(equipname)
 SUBPARTITIONS 8 STORE IN (ts1, ts2, ts3, ts4)
 (PARTITION p1 VALUES LESS THAN (1000),
  PARTITION p2 VALUES LESS THAN (2000),
  PARTITION p3 VALUES LESS THAN (3000),
  PARTITION p4 VALUES LESS THAN (MAXVALUE));

----------------------------------------------------------------
/* Range-List Partitioning */

CREATE TABLE regional_sales
 (ticket_no NUMBER,
 sale_year INT NOT NULL,
 sale_month INT NOT NULL,
 sale_day DATE,
 destination_city CHAR(3),
 start_city CHAR(3))
 PARTITION BY RANGE(sale_day)
 SUBPARTITION BY LIST (start_city)
 (PARTITION q1_2004 VALUES LESS THAN (TO_DATE('1-APR-2004','DD-MON-YYYY'))
 TABLESPACE ts1
 (SUBPARTITION q12004_northeast_sales VALUES ('NYC','BOS','PEN'),
 SUBPARTITION q12004_southwest_sales VALUES ('DFW','ORL','HOU'),
 SUBPARTITION q12004_pacificwest_sales VALUES ('SAN','LOS','WAS'),
 SUBPARTITION q12004_southeast_sales VALUES ('MIA','CHA','ATL')
 ),
 PARTITION q2_2004 VALUES LESS THAN (TO_DATE('1-JUL-2004','DD-MON-YYYY'))
 TABLESPACE ts2
 (SUBPARTITION q22004_northeast_sales VALUES ('NYC','BOS','PEN'),
 SUBPARTITION q22004_southwest_sales VALUES ('DFW','ORL','HOU'),
 SUBPARTITION q22004_pacificwest_sales VALUES ('SAN','LOS','WAS'),
 SUBPARTITION q22004_southeast_sales VALUES ('MIA','CHA','ATL')
 ),
 PARTITION q3_2004 VALUES LESS THAN (TO_DATE('1-OCT-2004','DD-MON-YYYY'))
 TABLESPACE ts3
 (SUBPARTITION q32004_northeast_sales VALUES ('NYC','BOS','PEN'),
 SUBPARTITION q32004_southwest_sales VALUES ('DFW','ORL','HOU'),
 SUBPARTITION q32004_pacificwest_sales VALUES ('SAN','LOS','WAS'),
 SUBPARTITION q32004_southeast_sales VALUES ('MIA','CHA','ATL')
 ),
 PARTITION q4_2004 VALUES LESS THAN (TO_DATE('1-JAN-2005','DD-MON-YYYY'))
 TABLESPACE ts4
 (SUBPARTITION q42004_northeast_sales VALUES ('NYC','BOS','PEN'),
 SUBPARTITION q42004_southwest_sales VALUES ('DFW','ORL','HOU'),
 SUBPARTITION q42004_pacificwest_sales VALUES ('SAN','LOS','WAS'),
 SUBPARTITION q42004_southeast_sales VALUES ('MIA','CHA','ATL')
 )
 );
----------------------------------------------------------------

/* Interval-Range Partitioned Tables or Range-Range */

CREATE TABLE sales
( prod_id NUMBER(6),
  cust_id NUMBER,
  time_id DATE,
  channel_id CHAR(1),
  promo_id NUMBER(6),
  quantity_sold NUMBER(3),
  amount_sold NUMBER(10,2)
)
PARTITION BY RANGE (time_id) INTERVAL (NUMTODSINTERVAL(1,'DAY'))
SUBPARTITION BY RANGE(amount_sold)
```

```
 SUBPARTITION TEMPLATE
 ( SUBPARTITION p_low VALUES LESS THAN (1000),
   SUBPARTITION p_medium VALUES LESS THAN (4000),
   SUBPARTITION p_high VALUES LESS THAN (8000),
   SUBPARTITION p_ultimate VALUES LESS THAN (maxvalue))
( PARTITION before_2000 VALUES LESS THAN (TO_DATE('01-JAN-2000','dd-MON-yyyy')));

---------------------------------------------------------------

/* Interval-List Partitioned Tables */

CREATE TABLE sales
( prod_id NUMBER(6)
, cust_id NUMBER
, time_id DATE
, channel_id CHAR(1)
, promo_id NUMBER(6)
, quantity_sold NUMBER(3)
, amount_sold NUMBER(10,2)
)
PARTITION BY RANGE (time_id) INTERVAL (NUMTODSINTERVAL(1,'DAY'))
SUBPARTITION BY LIST (channel_id)
SUBPARTITION TEMPLATE
(SUBPARTITION p_catalog VALUES ('C')
 , SUBPARTITION p_internet VALUES ('I')
 , SUBPARTITION p_partners VALUES ('P')
 , SUBPARTITION p_direct_sales VALUES ('S')
 , SUBPARTITION p_tele_sales VALUES ('T'))
( PARTITION before_2000 VALUES LESS THAN (TO_DATE('01-JAN-2000','dd-MON-yyyy')));
```

Partition Maintenance Operations

- Range and list partitions can't be coalesced.
- Hash partitions can't be dropped, split, or merged.
- Only list partitions allow the modification of partitions by adding and dropping the partition values.

```
-- Adding Partitions
ALTER TABLE ticket_sales
ADD PARTITION sales_quarter5 VALUES LESS THAN
 (TO_DATE('1-APR-2005','DD-MON-YYYY')) TABLESPACE ts5;

-- Splitting a Partition
ALTER TABLE ticket_sales
 SPLIT PARTITION ticket_sales01 AT (2000) INTO
 (PARTITION ticket_sales01A, ticket_sales01B);

-- Merging Partitions
ALTER TABLE ticket_sales
 MERGE PARTITIONS ticket_sales01, ticket_sales02 INTO PARTITION ticket_sales02;

-- Renaming Partitions
ALTER TABLE RENAME PARTITION fight_sales01 TO quarterly_sales01;

-- Exchanging Partitions
-- convert a regular nonpartitioned table into a partition of a partitioned table
-- it only renames the source and destination
ALTER TABLE ticket_sales
EXCHANGE PARTITION ticket_sales02 WITH ticket_sales03;

-- Dropping Partitions
```

```
-- use UPDATE GLOBAL INDEXES keyword if data exists in the partition
ALTER TABLE ticket_sales DROP PARTITION ticket_sales01;

-- Coalescing Partitions
-- applied on hash-partitioned and list-partitioned
--  to reduce number of partitions
ALTER TABLE ticket_sales COALESCE PARTITION;
```

Setting Deferred Segment Creation

- When the table (heap or partitioned) is created, no segment is assigned until a row is inserted.

```
(Oracle 11g)
-- db level:
show parameter DEFERRED_SEGMENT_CREATION
alter system set deferred_segment_creation=true scope=both ;


-- table level
create table test ( .. ) SEGMENT CREATION DEFERRED partition by .. ;
create table test ( .. ) SEGMENT CREATION IMMEDIATE ;

-- (11.2.0.2) If you want to create the segments for objects where
--  SEGMENT CREATION DEFERRED is set without waiting for
--  inserting any rows:

-- all the objects in a schema:
conn / as sysdba
begin
 DBMS_SPACE_ADMIN.MATERIALIZE_DEFERRED_SEGMENTS (
 schema_name =>'SA');
end;

-- specific table:
begin
 DBMS_SPACE_ADMIN.MATERIALIZE_DEFERRED_SEGMENTS (
 schema_name =>'SA', table_name=>'EMP');
end;

-- specific partition
begin
 DBMS_SPACE_ADMIN.MATERIALIZE_DEFERRED_SEGMENTS (
  schema_name =>'SA', table_name=>'EMP',
  partition_name=>'PAR01');
end;
```

Creating Multiple Tables and Views in a Single Operation

- If one included statement fails, non of the other statements will be executed.

```
 CREATE SCHEMA AUTHORIZATION scott
 CREATE TABLE dept (
  deptno NUMBER(3,0) PRIMARY KEY,
  dname VARCHAR2(15),
  loc VARCHAR2(25))
 CREATE TABLE emp (
  empno NUMBER(5,0) PRIMARY KEY,
  ename VARCHAR2(15) NOT NULL,
  job VARCHAR2(10),
  mgr NUMBER(5,0),
```

```
  hiredate DATE DEFAULT (sysdate),
  sal NUMBER(7,2),
  comm NUMBER(7,2),
  deptno NUMBER(3,0) NOT NULL
  CONSTRAINT dept_fkey REFERENCES dept)
CREATE VIEW sales_staff AS
  SELECT empno, ename, sal, comm
  FROM emp
  WHERE deptno = 30
  WITH CHECK OPTION CONSTRAINT sales_staff_cnst
GRANT SELECT ON sales_staff TO human_resources;
```

Collecting Object Statistics

Collecting Index Statistics

- granularity parameter takes:
  - 'DEFAULT' - gathers global and partition-level statistics
  - 'SUBPARTITION' - gathers subpartition-level statistics
  - 'PARTITION' - gathers partition-level statistics
  - 'GLOBAL' - gathers global statistics
  - 'ALL' - gathers all (subpartition, partition, and global)

```
begin
DBMS_STATS.GATHER_INDEX_STATS (
 ownname =>'HR',
 indname =>'EMP_DEPARTMENT_IX',
 partname  =>'', -- DEFAULT NULL
 estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -- 0.1 to 100 DEFAULT
to_estimate_percent_type (GET_PARAM('ESTIMATE_PERCENT'))
 stattab  => null, -- DEFAULT NULL
 statid  => null, -- DEFAULT NULL
 statown => null, --  DEFAULT NULL
 degree  => null, -- parallelism NUMBER DEFAULT
to_degree_type(get_param('DEGREE'))
 granularity => 'ALL', -- DEFAULT GET_PARAM('GRANULARITY')
 no_invalidate => false, -- BOOLEAN DEFAULT to_no_invalidate_type
(GET_PARAM('NO_INVALIDATE')),
 force => false -- BOOLEAN DEFAULT FALSE
 );
end;
/
```

Collecting Table Statistics

- method_opt Accepts:
  - FOR ALL [INDEXED | HIDDEN] COLUMNS [size_clause]
  - FOR COLUMNS [size clause] column|attribute [size_clause] [,column|attribute [size_clause]...]
  - size_clause is defined as size_clause := SIZE {integer | REPEAT | AUTO | SKEWONLY}

- integer : Number of histogram buckets. Must be in the range [1,254].

- REPEAT : Collects histograms only on the columns that already have histograms.

- AUTO : Oracle determines the columns to collect histograms based on data distribution and the workload of the columns.

- SKEWONLY : Oracle determines the columns to collect histograms based on the data distribution of the columns.

The default is FOR ALL COLUMNS SIZE AUTO

- granularity takes: 'ALL', 'AUTO', 'DEFAULT', 'GLOBAL', 'GLOBAL AND PARTITION', 'PARTITION', 'SUBPARTITION'

```
begin
DBMS_STATS.GATHER_TABLE_STATS (
 ownname =>'HR',
 tabname =>'NAMES',
 partname  =>'',
 estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -- NUMBER DEFAULT
to_estimate_percent_type(get_param('ESTIMATE_PERCENT'))
 block_sample => false, -- DEFAULT FALSE
 method_opt  =>'FOR ALL COLUMNS SIZE AUTO' , -- DEFAULT
get_param('METHOD_OPT')
 degree =>1, -- DEFAULT to_degree_type(get_param('DEGREE'))
 granularity =>'ALL' , -- DEFAULT GET_PARAM('GRANULARITY')
 cascade =>true, -- to_cascade_type(get_param('CASCADE'))
 stattab =>'',
 statid  =>'',
 statown =>'',
 no_invalidate => false, -- DEFAULT to_no_invalidate_type
(get_param('NO_INVALIDATE'))
 force => false
);
end;
/
```

Collecting Schema Statistics

- method_opt Accepts:
  - FOR ALL [INDEXED | HIDDEN] COLUMNS [size_clause]
  - FOR COLUMNS [size clause] column|attribute [size_clause] [,column|attribute [size_clause]...]
  - size_clause is defined as size_clause := SIZE {integer | REPEAT | AUTO | SKEWONLY}

- integer : Number of histogram buckets. Must be in the range [1,254].

- REPEAT : Collects histograms only on the columns that already have histograms.

- AUTO : Oracle determines the columns to collect histograms based on data distribution and the workload of the columns.

- SKEWONLY : Oracle determines the columns to collect histograms based on the data distribution of the columns.

The default is FOR ALL COLUMNS SIZE AUTO

- granularity takes: 'ALL', 'AUTO', 'DEFAULT', 'GLOBAL', 'GLOBAL AND PARTITION', 'PARTITION', 'SUBPARTITION'
- options takes : GATHER, GATHER AUTO, GATHER STALE, GATHER EMPTY. It also takes LIST STALE, LIST EMPTY, LIST AUTO (you should use an overloaded procedure different from the example above).

Note: GET_PARAM('..')is used till version 10g. In 11g, DBMS_STATS.GET_PREFS should be used.

```
begin
DBMS_STATS.GATHER_SCHEMA_STATS (
 ownname =>'HR',
```

```
 estimate_percent =>DBMS_STATS.AUTO_SAMPLE_SIZE, --0.1-100 DEFAULT
to_estimate_percent_type(get_param('ESTIMATE_PERCENT'))
 block_sample =>FALSE,
 method_opt =>'FOR ALL COLUMNS SIZE AUTO', -- DEFAULT get_param('METHOD_OPT')
 degree =>NULL, -- to_degree_type(get_param('DEGREE')),
 granularity =>'ALL', --  DEFAULT GET_PARAM('GRANULARITY'),
 cascade =>TRUE, -- DEFAULT to_cascade_type(get_param('CASCADE')),
 stattab =>'', --  DEFAULT NULL,
 statid =>'', --  DEFAULT NULL,
 options =>'GATHER', --  DEFAULT 'GATHER',
 statown =>'', --  DEFAULT NULL,
 no_invalidate =>false,-- DEFAULT to_no_invalidate_type
(get_param('NO_INVALIDATE')
 force => FALSE);
end;
/


begin
DBMS_STATS.GATHER_SCHEMA_STATS (
 ownname =>'HR',
 estimate_percent =>NULL,
 method_opt =>'FOR ALL COLUMNS SIZE AUTO',
 degree => DBMS_STATS.DEFAULT_DEGREE,
 granularity =>'ALL',
 cascade =>TRUE,
 options =>'GATHER AUTO');
end;
/
```

Validating Tables, Indexes, Clusters, and Materialized Views

- Validating an object means verifying the integrity of the structure of it.
- If a corruption is reported, re-create the object.

```
ANALYZE TABLE emp VALIDATE STRUCTURE;
ANALYZE TABLE emp VALIDATE STRUCTURE CASCADE;
ANALYZE TABLE emp VALIDATE STRUCTURE CASCADE ONLINE;
ANALYZE INDEX loc_country_ix VALIDATE STRUCTURE;
```

Listing Chained and Migrated Rows of Tables and Clusters

```
-- set up
--  following script creates CHAINED_ROWS table
@ $ORACLE_HOME$\RDBMS\ADMIN\utlchain.sql

-- handling chained and migrated rows
-- 1
ANALYZE TABLE order_hist LIST CHAINED ROWS;
ANALYZE CLUSTER emp_dept LIST CHAINED ROWS INTO CHAINED_ROWS;
-- 2
SELECT * FROM CHAINED_ROWS
WHERE TABLE_NAME = 'ORDER_HIST';
-- 3 if you have too many rows
--  intermediate table to hold the chained rows
```

```
CREATE TABLE int_order_hist
 AS SELECT *
 FROM order_hist
 WHERE ROWID IN
 (SELECT HEAD_ROWID
  FROM CHAINED_ROWS
  WHERE TABLE_NAME = 'ORDER_HIST');
-- 4
DELETE FROM order_hist
 WHERE ROWID IN
 (SELECT HEAD_ROWID
  FROM CHAINED_ROWS
  WHERE TABLE_NAME = 'ORDER_HIST');
-- 5
INSERT INTO order_hist
 SELECT * FROM int_order_hist;
-- 6
DROP TABLE int_order_history;
-- 7
DELETE FROM CHAINED_ROWS WHERE TABLE_NAME = 'ORDER_HIST';
-- 8 Use the ANALYZE statement again, and query the output table. Any rows that
appear in the output table are chained. You can eliminate chained rows only by
increasing your data block size.
```

Truncating Tables or Clusters

```
 TRUNCATE TABLE emp_dept DROP STORAGE; -- default, leave space of MINIEXTENTS
 TRUNCATE TABLE emp_dept REUSE STORAGE;
 TRUNCATE TABLE emp DROP ALL STORAGE; -- all segments will be dropped
```

Enabling and Disabling Triggers

```
select table_name, trigger_name from user_triggers where status='DISABLED';
ALTER TRIGGER reorder ENABLE;
ALTER TRIGGER reorder DISABLE;
ALTER TABLE inventory ENABLE ALL TRIGGERS;
ALTER TABLE inventory DISABLE ALL TRIGGERS;
```

Managing Integrity Constraints

Setting Constraint States and Deferability

- Constraint states:

ENABLE, VALIDATE : inserts are checked, existing data in the table is checked
ENABLE, NOVALIDATE (fast): inserts are checked, existing data in the table is NOT checked
DISABLE, VALIDATE : DML on constrained columns disallowed
DISABLE, NOVALIDATE : inserts are NOT checked, existing data in the table is NOT checked

- Constraint Checking (Deferability)

Nondeferred: enforced at the end of every DML statement.
Deferred : enforced when a transaction is committed.

- see next section for altering a constraint

```
CONSTRAINT ... [[NOT] DEFERRABLE INITIALLY [DEFERRED|IMMEDIATE]] [ENABLE|DISABLE]
[VALIDATE|NOVALIDATE]

ALTER SESSION SET CONSTRAINT[S] = {IMMEDIATE|DEFERRED|DEFAULT}

-- takes effect in TRANSACTION level
--  the trans is rolled back, if COMMIT fails
--  you can use SET CONSTRAINTS ALL IMMEDIATE to see whether COMMIT will fail before it
--  really commits
SET CONSTRAINT | CONSTRAINTS  {constraint |ALL } {IMMEDIATE|DEFERRED}
```

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE,STATUS,DEFERRABLE,DEFERRED,VALIDATED
FROM USER_CONSTRAINTS
WHERE CONSTRAINT_NAME='';
ALTER TABLE emp ADD PRIMARY KEY (empno) DISABLE;
ALTER TABLE countries ADD (UNIQUE(country_name) ENABLE NOVALIDATE);
ALTER TABLE myTable ADD CONSTRAINT uq UNIQUE (id) DEFERRABLE INITIALLY
DEFERRED;
-- takes effect in TRANSACTION level
-- not deferrable constraints are NOT affected
set constraint job_fk deferred;
set constraint job_fk immediate;
set constraints all deferred;
```

Modifying, Renaming, or Dropping Existing Integrity Constraints

```
ALTER TABLE dept DISABLE CONSTRAINT dname_ukey;
ALTER TABLE dept DISABLE PRIMARY KEY KEEP INDEX, DISABLE UNIQUE (dname, loc)
KEEP INDEX;
-- all FKs are also disabled
ALTER TABLE dept DISABLE PRIMARY KEY CASCADE;
ALTER TABLE dept ENABLE NOVALIDATE CONSTRAINT dname_ukey;
ALTER TABLE dept ENABLE NOVALIDATE PRIMARY KEY, ENABLE NOVALIDATE UNIQUE
(dname, loc);
ALTER TABLE dept MODIFY CONSTRAINT dname_key VALIDATE;
ALTER TABLE dept MODIFY PRIMARY KEY ENABLE NOVALIDATE;

ALTER TABLE dept RENAME CONSTRAINT dname_ukey TO dname_unikey;
ALTER TABLE dept DROP UNIQUE (dname, loc);
ALTER TABLE emp  DROP PRIMARY KEY KEEP INDEX, DROP CONSTRAINT dept_fkey;
```

Reporting Constraint Exceptions

- The target is to put all the rows that voilate sepcific integrity contraints in the EXCEPTIONS table.

```
1. create the EXCEPTIONS table
ORACLE_HOME\RDBMS\ADMIN\utlexcpt.sql
2.
ALTER TABLE dept ENABLE PRIMARY KEY EXCEPTIONS INTO EXCEPTIONS;
3.
SELECT * FROM EXCEPTIONS;
4. join the master table with the EXCEPTIONS
SELECT deptno, dname, loc FROM dept, EXCEPTIONS
 WHERE EXCEPTIONS.constraint = 'SYS_C00610'
 AND dept.rowid = EXCEPTIONS.row_id;
5. fix the incorrect data
6.
DELETE FROM EXCEPTIONS WHERE constraint = 'SYS_C00610';
```

Obtaining Information on Constraints

```
select * from USER_CONSTRAINTS;
select * from USER_CONS_COLUMNS;
```

Renaming Schema Objects

```
rename mytable to mytable2
```

Managing Object Dependencies

Manually Recompiling Views Procedures and Packages

```
ALTER VIEW emp_dept COMPILE;
ALTER PROCEDURE update_salary COMPILE;
ALTER PACKAGE acct_mgmt COMPILE BODY;
ALTER PACKAGE acct_mgmt COMPILE PACKAGE;
```

Switching to a Different Schema

```
CONNECT scott/tiger
ALTER SESSION SET CURRENT_SCHEMA = joe;
-- the following is equivalent to SELECT * FROM joe.emp
SELECT * FROM emp;
```

Using DBMS_METADATA to Display Information About Schema Objects

```
SET LONG 9999
-- storage clauses are not to be returned
EXECUTE
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'STORAGE',f
alse);
-- fetch the DDL of all tables (relational and objects)
SELECT DBMS_METADATA.GET_DDL('TABLE',u.table_name)
 FROM USER_ALL_TABLES u
 WHERE u.nested='NO'
 AND (u.iot_type is null or u.iot_type='IOT');
-- return the parameter value to its default
EXECUTE
DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'DEFAULT');
```

Specifying Storage Parameters at Object Creation

- Objects include: Tables, Clusters, Indexes, LOBs, Varrays, Nested Tables, MViews

```
CREATE TABLE MYTABLE(COL1 NUMBER, COL2 VARCHAR2(100))
STORAGE (INITIAL 10M BUFFER_POOL DEFAULT); -- MINEXTENTS

SELECT EXTENT_ID, BYTES/1024 KB
FROM DBA_EXTENTS
WHERE SEGMENT_TYPE='TABLE' AND SEGMENT_NAME='MYTABLE';
```

```
CREATE TABLE ...
( .., CONSTRAINT UQ)CON UNIQUE (PK_id)
 USING INDEX
 TABLESPACE MYTBS
 STORAGE (INITIAL 8K ) );

CREATE MATERIALIZED VIEW ..
TABLESPACE MYTBS
STORAGE (INITIAL 50K)
USING INDEX STORAGE (INITIAL 25K)
...
```

Managing Resumable Space Allocation

Enabling Resumable Space Allocation

```
-- the user should have this priv
grant RESUMABLE to hr;

select value from v$parameter where upper(name)='RESUMABLE_TIMEOUT';

-- to enable resumable space allocation system wide
--  0 to disable
alter system set RESUMABLE_TIMEOUT = 3600;
-- or in session level
alter session set RESUMABLE_TIMEOUT = 3600;

-- beside the RESUMABLE_TIMEOUT parameter
ALTER SESSION ENABLE RESUMABLE;
ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600; -- default 7200
-- 4340 is SID. NULL for current session
EXEC DBMS_RESUMABLE.set_session_timeout(4340,18000);
ALTER SESSION DISABLE RESUMABLE;

-- naming resumable statements
-- default is 'User username(userid), Session sessionid, Instance instanceid'
--  name is used in the DBA_RESUMABLE and USER_RESUMABLE views
ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600 NAME 'insert into table';
```

Detecting Suspended Statements

- The AFTER SUSPEND System Event and Trigger is executed after a SQL statement has been suspended.

```
-- source: Oracle Documentation
CREATE OR REPLACE TRIGGER resumable_default
AFTER SUSPEND ON DATABASE
DECLARE
/* declare transaction in this trigger is autonomous */
/* this is not required because transactions within a trigger are always
autonomous */
 PRAGMA AUTONOMOUS_TRANSACTION;
 cur_sid NUMBER;
 cur_inst NUMBER;
 errno NUMBER;
```

```
 err_type VARCHAR2;
 object_owner VARCHAR2;
 object_type VARCHAR2;
 table_space_name VARCHAR2;
 object_name VARCHAR2;
 sub_object_name VARCHAR2;
 error_txt VARCHAR2;
 msg_body VARCHAR2;
 ret_value BOOLEAN;
 mail_conn UTL_SMTP.CONNECTION;
BEGIN
 -- Get session ID
 SELECT DISTINCT(SID) INTO cur_SID FROM V$MYSTAT;

 -- Get instance number
 cur_inst := userenv('instance');

 -- Get space error information
 ret_value :=
DBMS_RESUMABLE.SPACE_ERROR_INFO(err_type,object_type,object_owner,table_space
_name,object_name, sub_object_name);

/*
-- If the error is related to undo segments, log error, send email
-- to DBA, and abort the statement. Otherwise, set timeout to 8 hours.
--
-- sys.rbs_error is a table which is to be
-- created by a DBA manually and defined as
-- (sql_text VARCHAR2(1000), error_msg VARCHAR2(4000), suspend_time DATE)
*/

IF OBJECT_TYPE = 'UNDO SEGMENT' THEN
 /* LOG ERROR */
 INSERT INTO sys.rbs_error (
 SELECT SQL_TEXT, ERROR_MSG, SUSPEND_TIME
 FROM DBA_RESUMABLE
 WHERE SESSION_ID = cur_sid AND INSTANCE_ID = cur_inst);

 SELECT ERROR_MSG INTO error_txt FROM DBMS_RESUMABLE
 WHERE SESSION_ID = cur_sid and INSTANCE_ID = cur_inst;

 -- Send email to receipient via UTL_SMTP package
 msg_body:='Subject: Space Error Occurred
 Space limit reached for undo segment ' || object_name ||
 on ' || TO_CHAR(SYSDATE, 'Month dd, YYYY, HH:MIam') ||
 '. Error message was ' || error_txt;
 mail_conn := UTL_SMTP.OPEN_CONNECTION('localhost', 25);
 UTL_SMTP.HELO(mail_conn, 'localhost');
 UTL_SMTP.MAIL(mail_conn, 'sender@localhost');
 UTL_SMTP.RCPT(mail_conn, 'recipient@localhost');
 UTL_SMTP.DATA(mail_conn, msg_body);
 UTL_SMTP.QUIT(mail_conn);
 -- Abort the statement
 DBMS_RESUMABLE.ABORT(cur_sid);
ELSE
 -- Set timeout to 8 hours
 DBMS_RESUMABLE.SET_TIMEOUT(28800);
```

```
 END IF;

 /* commit autonomous transaction */
 COMMIT;
 END;
 /
```

Obtaining Information about Suspended Statements

```
 SELECT USER_ID,SESSION_ID,INSTANCE_ID,
  COORD_INSTANCE_ID,COORD_SESSION_ID,STATUS,
  TIMEOUT,START_TIME,SUSPEND_TIME,
  RESUME_TIME,NAME,SQL_TEXT,
  ERROR_NUMBER,ERROR_PARAMETER1,ERROR_PARAMETER2,
  ERROR_PARAMETER3,ERROR_PARAMETER4,ERROR_PARAMETER5, ERROR_MSG
 FROM DBA_RESUMABLE;

 SELECT SID,SEQ#,EVENT,
  P1TEXT,P1,P1RAW,
  P2TEXT,P2,P2RAW,
  P3TEXT,P3,P3RAW,
  WAIT_CLASS_ID,WAIT_CLASS#,WAIT_CLASS,
  WAIT_TIME,SECONDS_IN_WAIT,STATE
 FROM V$SESSION_WAIT
 WHERE EVENT LIKE '%statement suspended%'

 exec DBMS_RESUMABLE.ABORT(159);
 select DBMS_RESUMABLE.GET_SESSION_TIMEOUT(159) from dual;
 exec DBMS_RESUMABLE.SET_SESSION_TIMEOUT(sessionID=>159,timeout=>8000);
```

Reclaiming Wasted Space

Displaying Information About Space Usage for Schema Objects

```
 /* Using PL SQL */
 SET SERVEROUTPUT ON
 declare
 -- IN vars
  v_segment_owner VARCHAR2(30) :='HR';
  v_segment_name  VARCHAR2(30) :='NAMES';
  v_segment_type  VARCHAR2(30) :='TABLE'; -- TABLE, TABLE PARTITION, TABLE
SUBPARTITION, INDEX, INDEX PARTITION, INDEX SUBPARTITION, CLUSTER, LOB, LOB
PARTITION, LOB SUBPARTITION
  v_partition_name VARCHAR2(30) :=NULL;
 -- OUT vars
  v_total_blocks NUMBER;
  v_total_bytes  NUMBER;
  v_unused_blocks  NUMBER;
  v_unused_bytes NUMBER;
  v_last_used_extent_file_id NUMBER; -- file id of last Extend containing data
  v_filename varchar2(300);
  v_last_used_extent_block_id NUMBER;
  v_last_used_block NUMBER;
 begin
 DBMS_SPACE.UNUSED_SPACE (
  segment_owner => v_segment_owner,
  segment_name  => v_segment_name ,
  segment_type  => v_segment_type,
```

```
 total_blocks  => v_total_blocks ,
 total_bytes   => v_total_bytes ,
 unused_blocks => v_unused_blocks,
 unused_bytes  => v_unused_bytes ,
 last_used_extent_file_id  => v_last_used_extent_file_id ,
 last_used_extent_block_id =>v_last_used_extent_block_id ,
 last_used_block => v_last_used_block ,
 partition_name  => v_partition_name );

 if v_last_used_extent_file_id is not null then
  select file_name into v_filename from dba_data_files where file_id =
v_last_used_extent_file_id ;
 end if;

 DBMS_OUTPUT.PUT_LINE('Total Blocks : '||v_total_blocks);
 DBMS_OUTPUT.PUT_LINE('Total MBytes : '||to_char(v_total_bytes/1024/1024));
 DBMS_OUTPUT.PUT_LINE('Unused Blocks: '||v_unused_blocks);
 DBMS_OUTPUT.PUT_LINE('Unused MBytes: '||to_char(v_unused_bytes/1024/1024));
 DBMS_OUTPUT.PUT_LINE('File of Last Extent Containing Data:
'||'('||v_last_used_extent_file_id  ||') '||v_filename );
 DBMS_OUTPUT.PUT_LINE('Starting Block ID of the Last Extent Containing Data:
'||v_last_used_block );
 DBMS_OUTPUT.PUT_LINE('Last Block Within This Extent Which Contains Data:
'||v_last_used_block );
end;
/

-- space usage of data blocks under the segment High Water Mark with
--  auto segment space management Tablespaces
-- following code in Oracle 10g
set serveroutput on

declare
 -- IN vars
 v_segment_owner VARCHAR2(100):='HR';
 v_segment_name VARCHAR2(100) :='NAMES2';
 v_segment_type VARCHAR2(100) :='TABLE';
 v_partition_name VARCHAR2(100) :=null;

 -- OUT vars
 v_unformatted_blocks   NUMBER;
 v_unformatted_bytes   NUMBER;
 v_fs1_blocks   NUMBER;
 v_fs1_bytes   NUMBER;
 v_fs2_blocks   NUMBER;
 v_fs2_bytes   NUMBER;
 v_fs3_blocks   NUMBER;
 v_fs3_bytes   NUMBER;
 v_fs4_blocks   NUMBER;
 v_fs4_bytes   NUMBER;
 v_full_blocks   NUMBER;
 v_full_bytes   NUMBER;
 v_segment_size_blocks NUMBER;
 v_segment_size_bytes NUMBER;
 v_used_blocks NUMBER;
 v_used_bytes NUMBER;
 v_expired_blocks NUMBER;
 v_expired_bytes NUMBER;
```

```
 v_unexpired_blocks NUMBER;
 v_unexpired_bytes NUMBER;
begin
DBMS_SPACE.SPACE_USAGE(
 segment_owner =>      v_segment_owner ,
 segment_name =>       v_segment_name ,
 segment_type =>       v_segment_type ,
 unformatted_blocks => v_unformatted_blocks ,
 unformatted_bytes =>  v_unformatted_bytes ,
 fs1_blocks => v_fs1_blocks ,
 fs1_bytes =>  v_fs1_bytes ,
 fs2_blocks => v_fs2_blocks ,
 fs2_bytes =>  v_fs2_bytes ,
 fs3_blocks => v_fs3_blocks ,
 fs3_bytes =>  v_fs3_bytes ,
 fs4_blocks => v_fs4_blocks ,
 fs4_bytes =>  v_fs4_bytes ,
 full_blocks =>        v_full_blocks ,
 full_bytes => v_full_bytes ,
 partition_name =>     v_partition_name );

 DBMS_OUTPUT.PUT_LINE('Free Space in MB:');
 DBMS_OUTPUT.PUT_LINE('Unformatted Mbytes: '||v_unformatted_bytes/1024/1024);
 DBMS_OUTPUT.PUT_LINE('Free Space in MB:');
 DBMS_OUTPUT.PUT_LINE(' 0  to 25%  free space: '|| v_fs1_bytes/1024/1024);
 DBMS_OUTPUT.PUT_LINE(' 25 to 50%  free space: '|| v_fs2_bytes/1024/1024);
 DBMS_OUTPUT.PUT_LINE(' 50 to 75%  free space: '|| v_fs3_bytes/1024/1024);
 DBMS_OUTPUT.PUT_LINE(' 75 to 100% free space: '|| v_fs4_bytes/1024/1024);
 DBMS_OUTPUT.PUT_LINE(' Full Mbytes          : '|| v_full_bytes/1024/1024);
 DBMS_OUTPUT.PUT_LINE('Free Space in Blocks:');
 DBMS_OUTPUT.PUT_LINE('Unformatted Blocks: ' || v_unformatted_blocks);
 DBMS_OUTPUT.PUT_LINE(' 0  to 25%  free space: '|| v_full_blocks);
 DBMS_OUTPUT.PUT_LINE(' 25 to 50%  free space: '|| v_fs2_blocks);
 DBMS_OUTPUT.PUT_LINE(' 50 to 75%  free space: '|| v_fs3_blocks);
 DBMS_OUTPUT.PUT_LINE(' 75 to 100% free space: '|| v_fs4_blocks);
 DBMS_OUTPUT.PUT_LINE(' Full blocks          : '|| v_full_blocks);

exception
 when others then
 if sqlcode = '-942' then
  DBMS_OUTPUT.PUT_LINE('Object Does not Exist.');
 else
  raise;
 end if;
end;
/


/* Using Views */
-- segment info
select OWNER,SEGMENT_NAME,PARTITION_NAME,SEGMENT_TYPE,S.TABLESPACE_NAME,
  HEADER_FILE,HEADER_BLOCK,S.BYTES/1024/1024 SEGMENT_SIZE_MB,
  D.BLOCKS SEGMENT_BLOCKS,EXTENTS,S.INITIAL_EXTENT,
S.NEXT_EXTENT NEXT_EXTENT_SIZE, S.MIN_EXTENTS,S.MAX_EXTENTS,S.PCT_INCREASE,
  FREELISTS,FREELIST_GROUPS,D.RELATIVE_FNO, D.FILE_NAME
from DBA_SEGMENTS S, DBA_TABLESPACES T, DBA_DATA_FILES D
where S.TABLESPACE_NAME = T.TABLESPACE_NAME AND T.TABLESPACE_NAME =
```

```
       D.TABLESPACE_NAME
   AND S.RELATIVE_FNO = D.RELATIVE_FNO
   AND T.TABLESPACE_NAME NOT IN ('SYSAUX','SYSTEM')
 order BY S.TABLESPACE_NAME;

-- extent info
SELECT EXTENT_ID, E.BLOCK_ID,E.BYTES/1024 EXTENT_SIZE_KB,
E.OWNER,E.SEGMENT_NAME,E.PARTITION_NAME,
  E.SEGMENT_TYPE,D.FILE_NAME, S.TABLESPACE_NAME,HEADER_FILE,
  HEADER_BLOCK SEG_HEADER_BLOCK,S.BYTES/1024/1024 SEGMENT_SIZE_MB,D.BLOCKS
SEGMENT_BLOCKS, EXTENTS SEG_EXTENTS
FROM DBA_EXTENTS E, DBA_SEGMENTS S, DBA_DATA_FILES D
WHERE E.OWNER=S.OWNER AND E.SEGMENT_NAME = S.SEGMENT_NAME
  AND NVL(E.PARTITION_NAME,'0') = NVL(S.PARTITION_NAME,'0')
  AND E.SEGMENT_TYPE = S.SEGMENT_TYPE
  AND E.FILE_ID = D.FILE_ID
  AND S.TABLESPACE_NAME NOT IN ('SYSAUX','SYSTEM')
ORDER BY E.SEGMENT_NAME, E.OWNER, E.PARTITION_NAME,E.EXTENT_ID;

-- Segment that cannot allocate additional extents
SELECT a.SEGMENT_NAME, a.SEGMENT_TYPE, a.TABLESPACE_NAME, a.OWNER
FROM DBA_SEGMENTS a
WHERE a.NEXT_EXTENT >= (SELECT MAX(b.BYTES)
            FROM DBA_FREE_SPACE b
            WHERE b.TABLESPACE_NAME = a.TABLESPACE_NAME)
 OR a.EXTENTS = a.MAX_EXTENTS
 OR a.EXTENTS = '8192' ; -- data_block_size

-- free extents within tablespaces
SELECT F.TABLESPACE_NAME,F.FILE_ID,F.BLOCK_ID,
  F.BYTES/1024/1024 FREE_MB, D.FILE_NAME
FROM DBA_FREE_SPACE F, DBA_DATA_FILES D
WHERE F.FILE_ID = D.FILE_ID
UNION
SELECT F.TABLESPACE_NAME, TO_NUMBER('') AS FILE_ID, TO_NUMBER('') AS BLOCK_ID,
  SUM(F.BYTES/1024/1024) FREE_MB, TO_CHAR('') AS FILE_NAME
FROM DBA_FREE_SPACE F
GROUP BY F.TABLESPACE_NAME, TO_NUMBER('') , TO_NUMBER('') ,TO_CHAR('')
ORDER BY TABLESPACE_NAME;
```

Segment Advisor

- The Automatic Segment Advisor is started by a Scheduler job that is configured to run during the default maintenance window(s).
- Input for DBMS_ADVISOR.CREATE_OBJECT

```
ATTR4 is unused. Specify NULL
OBJECT_TYPE            ATTR1           ATTR2           ATTR3
TABLESPACE             tbs name
TABLE          schema name     table name
INDEX          schema name     index name
TABLE PARTITION        schema name     table name      table partition name
INDEX PARTITION        schema name     index name      index partition name
TABLE SUBPART. schema name     table name      table subpartition name
INDEX SUBPART.         schema name     index name      index subpartition name
LOB                    schema name     segment name
LOB PARTITION  schema name     segment name    lob partition name
LOB SUBPART.           schema name     segment name    lob subpartition name
```

```
 -- object priv EXECUTE ON DBMS_ADVISOR or the system priv ADVISOR
```

```
-- run Segment Advisor on hr.employees
variable id number;
declare
 v_name varchar2(100);
 v_descr varchar2(500);
 obj_id number;
begin
 v_name:='Manual_Employees';
 v_descr:='Segment Advisor Example';
 dbms_advisor.create_task (
  advisor_name => 'Segment Advisor',
  task_id => :id,
  task_name => v_name,
  task_desc => v_descr);

 -- identify the target object
 dbms_advisor.create_object (
  task_name => v_name,
  object_type => 'TABLE',
  attr1 => 'HR',
  attr2 => 'EMPLOYEES',
  attr3 => NULL,
  attr4 => NULL,
  attr5 => NULL,
  object_id => obj_id);

 dbms_advisor.set_task_parameter(
  task_name => v_name,
  parameter => 'recommend_all',
  value => 'TRUE');
  dbms_advisor.execute_task(v_name);
end;
/

-- viewing segment results
-- asa_recommendations ( all_runs    in varchar2 DEFAULT 'TRUE',
--                       show_manual in varchar2 DEFAULT 'TRUE',
--                       show_findings in varchar2 DEFAULT 'FALSE' )
select tablespace_name, segment_name, segment_type, partition_name,
recommendations, c1 from
table(dbms_space.asa_recommendations('TRUE', 'TRUE', 'TRUE'));

select * from table(dbms_space.asa_recommendations('TRUE', 'TRUE', 'TRUE'));

-- alternatively use DBA_ADVISOR_*
SELECT OWNER,REC_ID,TASK_ID,
 TASK_NAME,FINDING_ID,TYPE,
 RANK,PARENT_REC_IDS,BENEFIT_TYPE,
 BENEFIT,ANNOTATION_STATUS,FLAGS
FROM DBA_ADVISOR_RECOMMENDATIONS
WHERE TASK_ID = :ID

SELECT OWNER,TASK_ID,TASK_NAME,
 FINDING_ID,TYPE,PARENT,
 OBJECT_ID,IMPACT_TYPE,IMPACT,MESSAGE,MORE_INFO
FROM DBA_ADVISOR_FINDINGS
WHERE TASK_ID = :ID
```

```
-- suggested actions to perform
SELECT OWNER,TASK_ID,TASK_NAME,
 REC_ID,ACTION_ID,OBJECT_ID,
 COMMAND,COMMAND_ID,FLAGS,
 ATTR1,ATTR2,ATTR3,
 ATTR4,ATTR5,ATTR6,
 NUM_ATTR1,NUM_ATTR2,NUM_ATTR3,
 NUM_ATTR4,NUM_ATTR5,MESSAGE
FROM DBA_ADVISOR_ACTIONS
WHERE TASK_ID = :ID

-- analyzed objects
SELECT OWNER,OBJECT_ID,TYPE,TYPE_ID,
 TASK_ID,TASK_NAME,ATTR1,ATTR2,
 ATTR3,ATTR4,ATTR5,OTHER
FROM DBA_ADVISOR_OBJECTS
WHERE TASK_ID = :ID

/*  information about the Advisor Runs */
SELECT AUTO_TASKID,SNAPID,SEGMENTS_SELECTED,SEGMENTS_PROCESSED,
 TABLESPACE_SELECTED,TABLESPACE_PROCESSED,RECOMMENDATIONS_COUNT,
 START_TIME, END_TIME
FROM DBA_AUTO_SEGADV_SUMMARY;

/* control information that show which object analyzed by the Advisor */
SELECT *
FROM DBA_AUTO_SEGADV_CTL
```

Shrinking Database Segments Online

```
-- required
ALTER TABLE names  ENABLE ROW MOVEMENT -- it acquires table lock

-- shrink table (defrage and then move the HWM)
ALTER TABLE names SHRINK SPACE; -- it acquires table lock

-- defrage and don't move HWM (later you can SHRINK SPACE)
ALTER TABLE names SHRINK SPACE COMPACT; -- no table lock

-- LOBs and the Indexes are alos shrinked
ALTER TABLE names SHRINK SPACE CASCADE;

-- shrink LOB only
ALTER TABLE names MODIFY LOB (perf_review) (SHRINK SPACE);

-- shrink partition
ALTER TABLE names MODIFY PARTITION cust_P1 SHRINK SPACE;

-- IOT
ALTER TABLE names SHRINK SPACE CASCADE;

-- IOT Overflow Segment
ALTER TABLE names OVERFLOW SHRINK SPACE;
```

Deallocating Unused Space

- deallocates unused space at the end of the segment; no data is moved

```
ALTER TABLE mytable DEALLOCATE UNUSED KEEP integer;
ALTER INDEX myindex DEALLOCATE UNUSED KEEP integer;
ALTER CLUSTER cluster DEALLOCATE UNUSED KEEP integer;
```

Capacity Planning for Database Objects

Estimating the Space Use of a Table

- DBMS_SPACE.CREATE_TABLE_COST can be used on dictionary-managed or locally managed tablespaces.

```
-- based on average row size
SET SERVEROUTPUT ON
declare
 v_table_name varchar2(30) :='NAMES';
 v_owner varchar2(30) :='HR';
 v_tablespace_name varchar2(100) := 'USERS';
 v_avg_row_size number ; -- anticipated average row size
 v_row_count number := 1000000;
 v_pct_free number := 10;
 v_used_bytes number;  -- User data space including PCTFREE
 v_alloc_bytes number; -- The anticipated size of the table taking into
account the tablespace extent characteristics
 x char(1);
begin

-- check the tablespace exists
begin
 SELECT 'x' into x from dba_tablespaces where
tablespace_name=v_tablespace_name ;
exception
 when no_data_found then
  raise_application_error(-20001,'Tablespace: ' || v_tablespace_name || '
does not exist');
end;

-- retreive average row size
begin
 SELECT AVG_ROW_LEN into V_avg_row_size FROM DBA_TABLES where
table_name=v_table_name and owner=v_owner;
exception
 when no_data_found then
  raise_application_error(-20002,'Table : ' || V_OWNER ||'.'||v_table_name ||
' does not exist');
end;

DBMS_SPACE.CREATE_TABLE_COST (
 tablespace_name => v_tablespace_name,
 avg_row_size => v_avg_row_size,
 row_count    =>  v_row_count,
 pct_free     =>  v_pct_free,
 used_bytes   =>  v_used_bytes,
 alloc_bytes  =>  v_alloc_bytes);

dbms_output.put_line('User Data Space (MB): '||
TO_CHAR(v_used_bytes/1024/1024 ,'999,999,999,999,999.99'));
dbms_output.put_line('Data Space in the Tablespace (MB): '||
```

```
TO_CHAR(v_alloc_bytes/1024/1024,'999,999,999,999,999.99'));
end;
/

-- based on column info
declare
 ub NUMBER;
 ab NUMBER;
 cl sys.create_table_cost_columns;
begin
cl := sys.create_table_cost_columns(
                sys.create_table_cost_colinfo('NUMBER',10),
                sys.create_table_cost_colinfo('VARCHAR2',30),
                sys.create_table_cost_colinfo('VARCHAR2',30),
                sys.create_table_cost_colinfo('DATE',NULL));
DBMS_SPACE.CREATE_TABLE_COST('SYSTEM',cl,100000,0,ub,ab);
DBMS_OUTPUT.PUT_LINE('Used Bytes: ' || TO_CHAR(ub));
DBMS_OUTPUT.PUT_LINE('Alloc Bytes: ' || TO_CHAR(ab));
end;
/
```

Obtaining Object Growth Trends

- User DBMS_SPACE.OBJECT_GROWTH_TREND
  - start_time Statistics generated after this time will be used in generating the growth trend
  - end_time Statistics generated until this time will be used in generating the growth trend
  - interval The interval at which to sample
  - skip_interpolated Whether interpolation of missing values should be skipped
  - single_data_point_flag Whether in the absence of statistics the segment should be sampled
- Returned columns:
  - timepoint The time at which the statistic was recorded
  - space_usage The space used by data
  - space_alloc The size of the segment including overhead and unused space
  - quality The quality of result: "GOOD", "INTERPOLATED", "PROJECTION"

    GOOD: accurate figure

    PROJECTED: figures projected from the data collected by the AWR

    INTERPOLATED: no data was available for calculations

```
COL TIMEPOINT FORMAT A30

SELECT *
FROM
table(
 DBMS_SPACE.OBJECT_GROWTH_TREND (
  object_owner =>'HR',
  object_name  =>'NAMES',
  object_type  =>'TABLE',
  partition_name =>NULL,
  start_time =>NULL,
  end_time   =>NULL,
  interval   =>to_dsinterval('0 00:10:00')  , -- Day to Second interval
  skip_interpolated => 'FALSE',
```

```
  timeout_seconds =>NULL,
  single_datapoint_flag =>'TRUE')
)
```

Using the SQL Access Advisor

Estimating the Space Use of a Table

- The SQL Access Advisor sources:
  - A hypothetical database workload
  - An actual workload you provide
  - SQL cache

```
-- Method 1: to invoke SQL Access Advisor task linked to a workload
declare
  l_taskname     VARCHAR2(30)   := 'sql_access_test_task';
  l_task_desc    VARCHAR2(128)  := 'Test SQL Access';
  l_wkld_name    VARCHAR2(30)   := 'test_work_load';
  l_saved_rows   NUMBER         := 0;
  l_failed_rows  NUMBER         := 0;
  l_num_found    NUMBER;
BEGIN
  -- create an SQL Access Advisor task.
  select COUNT(*)
  into   l_num_found
  from DBA_ADVISOR_TASKS
  where TASK_NAME =  l_taskname ;
  IF l_num_found = 0 THEN
   DBMS_ADVISOR.CREATE_TASK (
    ADVISOR_NAME => DBMS_ADVISOR.SQLACCESS_ADVISOR,
    TASK_NAME    => l_taskname,
    TASK_DESC    => l_task_desc);
  END IF;

  -- reset the task ( remove all recommendations, and intermediate data from
the task)
  DBMS_ADVISOR.RESET_TASK(TASK_NAME => l_taskname);

  -- create a workload.
  SELECT COUNT(*)
  INTO   l_num_found
  FROM   USER_ADVISOR_SQLW_SUM
  WHERE  WORKLOAD_NAME = l_wkld_name;

  IF l_num_found = 0 THEN
    DBMS_ADVISOR.CREATE_SQLWKLD(WORKLOAD_NAME => l_wkld_name);
  END IF;

  -- link the workload to the task
  SELECT count(*)
  INTO   l_num_found
  FROM   USER_ADVISOR_SQLA_WK_MAP
  WHERE  TASK_NAME     = l_taskname
    AND  WORKLOAD_NAME = l_wkld_name;

  IF l_num_found = 0 THEN
```

```
    DBMS_ADVISOR.ADD_SQLWKLD_REF(
      TASK_NAME     => l_taskname,
      WORKLOAD_NAME => l_wkld_name);
  END IF;

  -- Set workload parameters.
  DBMS_ADVISOR.SET_SQLWKLD_PARAMETER(l_wkld_name, 'ACTION_LIST',
DBMS_ADVISOR.ADVISOR_UNUSED);
  DBMS_ADVISOR.SET_SQLWKLD_PARAMETER(l_wkld_name, 'MODULE_LIST',
DBMS_ADVISOR.ADVISOR_UNUSED);
  DBMS_ADVISOR.SET_SQLWKLD_PARAMETER(l_wkld_name, 'SQL_LIMIT',
DBMS_ADVISOR.ADVISOR_UNLIMITED);
  DBMS_ADVISOR.SET_SQLWKLD_PARAMETER(l_wkld_name, 'ORDER_LIST',
'PRIORITY,OPTIMIZER_COST');
  DBMS_ADVISOR.SET_SQLWKLD_PARAMETER(l_wkld_name, 'USERNAME_LIST',
DBMS_ADVISOR.ADVISOR_UNUSED);
  DBMS_ADVISOR.SET_SQLWKLD_PARAMETER(l_wkld_name, 'VALID_TABLE_LIST',
DBMS_ADVISOR.ADVISOR_UNUSED);

 -- unmark the required option
 /*
  -- import the current contents of the server's SQL cache
  DBMS_ADVISOR.IMPORT_SQLWKLD_SQLCACHE(l_wkld_name, 'REPLACE', 2,
l_saved_rows, l_failed_rows);
  -- load a SQL workload from an existing SQL Tuning Set
  DBMS_ADVISOR.IMPORT_SQLWKLD_STS (l_wkld_name, 'SOURCE_STS_Name',
'REPLACE',2, l_saved_rows, l_failed_rows);
*/
  -- Set task parameters.
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, '_MARK_IMPLEMENTATION',
'FALSE');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'EXECUTION_TYPE',
'INDEX_ONLY');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'MODE', 'COMPREHENSIVE');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'STORAGE_CHANGE',
DBMS_ADVISOR.ADVISOR_UNLIMITED);
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DML_VOLATILITY', 'TRUE');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'ORDER_LIST',
'PRIORITY,OPTIMIZER_COST');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'WORKLOAD_SCOPE', 'PARTIAL');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DEF_INDEX_TABLESPACE',
DBMS_ADVISOR.ADVISOR_UNUSED);
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DEF_INDEX_OWNER',
DBMS_ADVISOR.ADVISOR_UNUSED);
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DEF_MVIEW_TABLESPACE',
DBMS_ADVISOR.ADVISOR_UNUSED);
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DEF_MVIEW_OWNER',
DBMS_ADVISOR.ADVISOR_UNUSED);

  -- Execute the task: control will not return till the execution finishes
  DBMS_ADVISOR.execute_task(task_name => l_taskname);
END;
/


-- Method 2: to invoke SQL Access Advisor linked to a specific STS
declare
  l_taskname     VARCHAR2(30)   := 'sql_access_test_task2';
  l_task_desc    VARCHAR2(128)  := 'Test SQL Access for a STS';
```

```
   l_wkld_name    VARCHAR2(30)   := 'test_work_load';
   l_sts_name     VARCHAR2(30)   := 'test_sts';
   l_saved_rows   NUMBER         := 0;
   l_failed_rows  NUMBER         := 0;
   l_num_found    NUMBER;
BEGIN
   -- create an SQL Access Advisor task, if it doesn't exist
   select COUNT(*)
   into   l_num_found
   from DBA_ADVISOR_TASKS
   where TASK_NAME =  l_taskname ;
  IF l_num_found = 0 THEN
  DBMS_ADVISOR.CREATE_TASK (
     ADVISOR_NAME => DBMS_ADVISOR.SQLACCESS_ADVISOR,
     TASK_NAME    => l_taskname,
     TASK_DESC    => l_task_desc);
   END IF;

   -- reset the task ( remove all recommendations, and intermediate data from
the task)
   DBMS_ADVISOR.RESET_TASK(TASK_NAME => l_taskname);

   -- check if STS already exists
   select count(*)
   into   l_num_found
   from   DBA_SQLSET where upper(NAME) = upper(l_sts_name) ;

   IF l_num_found <> 0 THEN
     DBMS_SQLTUNE.DROP_SQLSET ( sqlset_name => l_sts_name);
   END IF;

   -- create STS
   DBMS_SQLTUNE.CREATE_SQLSET(SQLSET_NAME => l_sts_name, DESCRIPTION =>'To
test Access Advisor');

  /* unmark the required option
  -- (Option 1) Load l_sts_name from an AWR baseline.
  -- The data has been filtered to select only the top 30 SQL statements
ordered by elapsed time.
  declare
   baseline_cur DBMS_SQLTUNE.SQLSET_CURSOR;
  begin
   -- a ref cursor is opened to select from the specified baseline
   OPEN baseline_cur FOR
     SELECT VALUE(p)
     FROM TABLE (DBMS_SQLTUNE.SELECT_WORKLOAD_REPOSITORY(
    'peak baseline',NULL, NULL, 'elapsed_time', NULL, NULL, NULL,30 )) p;

    -- Next the statements and their statistics are loaded into the STS
    DBMS_SQLTUNE.LOAD_SQLSET( SQLSET_NAME=>l_sts_name,
POPULATE_CURSOR=>baseline_cur);
  end;

  -- (Option 2) Load l_sts_name with SQL statements that are not owned by SYS
and
  --  their elapsed time is greater than 20,000 seconds.
  declare
    sql_cur DBMS_SQLTUNE.SQLSET_CURSOR;
```

```
 begin
   -- a ref cursor is opened to select the required SQL statments
   OPEN sql_cur FOR
     SELECT VALUE(p)
     FROM TABLE (DBMS_SQLTUNE.SELECT_CURSOR_CACHE('parsing_schema_name <>
''SYS'' and elapsed_time > 2000000',NULL, NULL, NULL, NULL,1, NULL, 'ALL'))
p;
     -- the statements are loaded into the STS
     DBMS_SQLTUNE.LOAD_SQLSET( SQLSET_NAME=>l_sts_name,
POPULATE_CURSOR=>sql_cur);
 end;

 -- (Option 3) Copy the contents of a SQL workload object to a SQL Tuning Set
 -- check the example above for creating a workload
  DBMS_ADVISOR.COPY_SQLWKLD_TO_STS ('My_WorkLoad', l_sts_name, 'REPLACE');
 */

  -- link the STS to the task
  DBMS_ADVISOR.ADD_STS_REF (l_taskname, null, l_sts_name);

  -- Set task parameters.
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, '_MARK_IMPLEMENTATION',
'FALSE');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'EXECUTION_TYPE',
'INDEX_ONLY');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'MODE', 'COMPREHENSIVE');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'STORAGE_CHANGE',
DBMS_ADVISOR.ADVISOR_UNLIMITED);
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DML_VOLATILITY', 'TRUE');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'ORDER_LIST',
'PRIORITY,OPTIMIZER_COST');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'WORKLOAD_SCOPE', 'PARTIAL');
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DEF_INDEX_TABLESPACE',
DBMS_ADVISOR.ADVISOR_UNUSED);
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DEF_INDEX_OWNER',
DBMS_ADVISOR.ADVISOR_UNUSED);
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DEF_MVIEW_TABLESPACE',
DBMS_ADVISOR.ADVISOR_UNUSED);
  DBMS_ADVISOR.SET_TASK_PARAMETER(l_taskname, 'DEF_MVIEW_OWNER',
DBMS_ADVISOR.ADVISOR_UNUSED);

  -- Execute the task: control will not return till the execution finishes
  DBMS_ADVISOR.EXECUTE_TASK(task_name => l_taskname);
END;
/


-- Method 3: Quick Tune for a single given statement
begin
  -- a task and a workload will be created then the task will be executed
  DBMS_ADVISOR.QUICK_TUNE(
    ADVISOR_NAME => DBMS_ADVISOR.SQLACCESS_ADVISOR,
    TASK_NAME    => 'names_quick_tune',
    ATTR1        => 'SELECT id FROM hr.names n WHERE id  = 100');
end;
/
```

Following are examples of how to manage the task and obtain information about the advisor's output repot:

```
-- while the task is being executed, you can monitor it using the following
query:
select TASK_NAME, STATUS, PCT_COMPLETION_TIME, ERROR_MESSAGE
from   DBA_ADVISOR_LOG
where  TASK_NAME ='sql_access_test_task';

-- if you need to terminate the executing task (may be time consuming)
exec DBMS_ADVISOR.CANCEL_TASK(TASK_NAME =>'sql_access_test_task');

-- Display the resulting script ( method 1)
SET LONG 100000
SET PAGESIZE 50000
SELECT DBMS_ADVISOR.GET_TASK_SCRIPT('sql_access_test_task') AS script FROM
dual;
SET PAGESIZE 24

-- Display the resulting script ( method 2)
CREATE DIRECTORY ADVISOR_RESULTS AS 'C:\TEMP\';
exec
DBMS_ADVISOR.CREATE_FILE(DBMS_ADVISOR.GET_TASK_SCRIPT('sql_access_test_task'),
'ADVISOR_RESULTS','advscript.sql');

-- alternatively, use the following queries
--  benefit is the total improvement in execution cost of all the queries
select REC_ID, RANK, BENEFIT, TYPE "Recommendation Type"
from   DBA_ADVISOR_RECOMMENDATIONS
where  TASK_NAME = 'sql_access_test_task'
order by RANK;

-- display processed statements in the workload
select SQL_ID, REC_ID, PRECOST, POSTCOST,
       (PRECOST-POSTCOST)*100/PRECOST AS PERCENT_BENEFIT
from   USER_ADVISOR_SQLA_WK_STMTS
where  TASK_NAME = 'sql_access_test_task' AND workload_name =
'test_work_load';

-- see the actions for each recommendations
select REC_ID, ACTION_ID, SUBSTR(COMMAND,1,30) AS COMMAND
from   USER_ADVISOR_ACTIONS where TASK_NAME = 'sql_access_test_task'
ORDER BY rec_id, action_id;

-- to delete a given task
exec DBMS_ADVISOR.DELETE_TASK('sql_access_test_task');
```

Managing Tables

Obtaining Information about Tables

```
-- 11g
SELECT OWNER,TABLE_NAME,TABLESPACE_NAME,CLUSTER_NAME,IOT_NAME,STATUS,PCT_FREE,
PCT_USED,INI_TRANS,MAX_TRANS,INITIAL_EXTENT,
NEXT_EXTENT,MIN_EXTENTS,MAX_EXTENTS,PCT_INCREASE,FREELISTS,FREELIST_GROUPS,LOGGING,
BACKED_UP,NUM_ROWS,BLOCKS,EMPTY_BLOCKS,AVG_SPACE,CHAIN_CNT,
AVG_ROW_LEN,AVG_SPACE_FREELIST_BLOCKS,NUM_FREELIST_BLOCKS,DEGREE,
INSTANCES,CACHE,TABLE_LOCK,SAMPLE_SIZE,LAST_ANALYZED,PARTITIONED,IOT_TYPE,
TEMPORARY,SECONDARY,NESTED,BUFFER_POOL,ROW_MOVEMENT,GLOBAL_STATS,USER_STATS,
DURATION,SKIP_CORRUPT,MONITORING,CLUSTER_OWNER,DEPENDENCIES,COMPRESSION,
COMPRESS_FOR,DROPPED,READ_ONLY
FROM DBA_TABLES
WHERE TABLE_NAME='EMP' AND OWNER='HR';

-- columns
SELECT
OWNER,TABLE_NAME,COLUMN_NAME,DATA_TYPE,DATA_TYPE_MOD,DATA_TYPE_OWNER,DATA_LENGTH,DATA_P
RECISION,DATA_SCALE,NULLABLE,COLUMN_ID,DEFAULT_LENGTH,DATA_DEFAULT,NUM_DISTINCT,LOW_VAL
UE,HIGH_VALUE,DENSITY,NUM_NULLS,NUM_BUCKETS,LAST_ANALYZED,SAMPLE_SIZE,CHARACTER_SET_NAM
E,CHAR_COL_DECL_LENGTH,GLOBAL_STATS,USER_STATS,AVG_COL_LEN,CHAR_LENGTH,CHAR_USED,V80_FM
T_IMAGE,DATA_UPGRADED,HISTOGRAM
FROM DBA_TAB_COLUMNS
WHERE TABLE_NAME='EMP' AND OWNER='HR';

-- table gathered statistics
SELECT
OWNER,TABLE_NAME,PARTITION_NAME,PARTITION_POSITION,SUBPARTITION_NAME,SUBPARTITION_POSIT
ION,OBJECT_TYPE,NUM_ROWS,BLOCKS,EMPTY_BLOCKS,AVG_SPACE,CHAIN_CNT,AVG_ROW_LEN,AVG_SPACE_
FREELIST_BLOCKS,NUM_FREELIST_BLOCKS,AVG_CACHED_BLOCKS,AVG_CACHE_HIT_RATIO,SAMPLE_SIZE,L
AST_ANALYZED,GLOBAL_STATS,USER_STATS,STATTYPE_LOCKED,STALE_STATS
FROM DBA_TAB_STATISTICS
WHERE TABLE_NAME='EMP' AND OWNER='HR';

-- comments
select * from DBA_TAB_COMMENTS;
select * from DBA_COL_COMMENTS;

-- object and relational tables
SELECT
OWNER,TABLE_NAME,TABLESPACE_NAME,CLUSTER_NAME,IOT_NAME,STATUS,PCT_FREE,PCT_USED,INI_TRA
NS,MAX_TRANS,INITIAL_EXTENT,NEXT_EXTENT,MIN_EXTENTS,MAX_EXTENTS,PCT_INCREASE,FREELISTS,
FREELIST_GROUPS,LOGGING,BACKED_UP,NUM_ROWS,BLOCKS,EMPTY_BLOCKS,AVG_SPACE,CHAIN_CNT,AVG_
ROW_LEN,AVG_SPACE_FREELIST_BLOCKS,NUM_FREELIST_BLOCKS,DEGREE,INSTANCES,CACHE,TABLE_LOCK
,SAMPLE_SIZE,LAST_ANALYZED,PARTITIONED,IOT_TYPE,OBJECT_ID_TYPE,TABLE_TYPE_OWNER,TABLE_T
YPE,TEMPORARY,SECONDARY,NESTED,BUFFER_POOL,ROW_MOVEMENT,GLOBAL_STATS,USER_STATS,DURATIO
N,SKIP_CORRUPT,MONITORING,CLUSTER_OWNER,DEPENDENCIES,COMPRESSION,COMPRESS_FOR,DROPPED
FROM DBA_ALL_TABLES
WHERE TABLE_NAME='EMP' AND OWNER='HR';
```

Creating Tables with some Options

```
-- specific tablspace
create table mytab ... tablspace mytbs
-- Parallelizing Table Creation
CREATE TABLE mytb PARALLEL AS SELECT ..
ALTER SESSION FORCE PARALLEL DDL; -- created tables will be parallelized

-- nologging
```

```
CREATE TABLE mytb NOLOGGING AS SELECT ..

-- compression
CREATE TABLE mytb ( ... ) COMPRESS FOR ALL OPERATIONS; -- OK in OLTP
CREATE TABLE sales_history ( ... ) COMPRESS FOR DIRECT_LOAD OPERATIONS; -- OK in OLAP
CREATE TABLE sales_history ( ... ) COMPRESS; -- same as above
CREATE TABLE sales
(saleskey number, quarter .. region varchar2(10)) COMPRESS
PARTITION BY LIST (region)
(PARTITION northwest VALUES ('NORTHWEST'),
 PARTITION southwest VALUES ('SOUTHWEST'),
 PARTITION northeast VALUES ('NORTHEAST') NOCOMPRESS,
 PARTITION southeast VALUES ('SOUTHEAST'),
 PARTITION western VALUES ('WESTERN'))

-- encrypting columns (REQUIRES OEPENED WALLET)
CREATE TABLE mytb ( .. ssn NUMBER(9) ENCRYPT, ..)

-- temporary table
-- transaction level
CREATE GLOBAL TEMPORARY TABLE mytb (..) ON COMMIT DELETE ROWS;
-- session level
CREATE GLOBAL TEMPORARY TABLE mytb (..) ON COMMIT PRESERVE ROWS;
```

DML Error Logging

- DML error logging works with INSERT, UPDATE, MERGE, and DELETE statements.
- The following conditions cause the statement to fail and roll back without invoking the error logging capability:
  - Violated deferred constraints
  - Out-of-space errors
  - Any direct-path INSERT operation (INSERT or MERGE) that raises a unique constraint or index violation
  - Any UPDATE or MERGE that raises a unique constraint or index violation

```
-- 1. Create an error logging table.
-- LONG, CLOB, BLOB, BFILE, and ADT datatypes are not supported in the
columns
EXECUTE DBMS_ERRLOG.CREATE_ERROR_LOG(dml_table_name=>'EMP',
err_log_table_name=>'ERR_EMP');
EXECUTE DBMS_ERRLOG.CREATE_ERROR_LOG(dml_table_name=>'EMP',
err_log_table_name=>'ERR_EMP', err_log_table_owner=>'HR',
err_log_table_space=>'USERS');
-- 2 include an error logging clause
INSERT INTO emp VALUES(1001, 'abcd', 75, NULL)
LOG ERRORS INTO ERR_EMP REJECT LIMIT UNLIMITED;
INSERT INTO hr.emp
SELECT .. FROM ..
LOG ERRORS INTO err_emp ('daily_load') REJECT LIMIT 25
-- 3 Query the error logging table and take corrective action
```

Enabling Direct-Path INSERT

- To activate direct-path INSERT in serial mode, you must specify the APPEND hint in each INSERT statement, either immediately after the INSERT keyword, or immediately after the SELECT keyword in the subquery of the INSERT statement.

```
insert /*+ append */ into customer (select xxx);
```

- When you are inserting in parallel DML mode, direct-path INSERT is the default. In order to run in parallel DML mode, the following requirements must be met:
  - You must have Oracle Enterprise Edition installed.
  - You must enable parallel DML in your session. To do this, run the following statement:

    ```
    ALTER SESSION { ENABLE | FORCE } PARALLEL DML;
    ```

  - You must specify the parallel attribute for the target table, either at create time or subsequently, or you must specify the PARALLEL hint for each insert operation.

    ```
    insert /*+ parallel */ into names3 (select * from names)
    ```

- If nologging is specified, the insert will finish faster.

```
select logging from dba_tables where owner='HR' and table_name='EMPLOYEES';
select logging from dba_tablespaces where tablespace_name = 'USERS';
```

- Index maintenance is done by end of command. Consider drop and then re-creating the indexes after the statement.
- Exclusive table lock is acquired.

Automatically Collecting Statistics on Tables

- In Oracle Database 11g, statistics are collected automatically for tables regardless of the the MONITORING and NOMONITORING keywords
  when creating them.

Altering Tables

- If you enable compression for all operations on a table, you can drop table columns. If you enable compression for direct-path inserts only, you cannot drop columns.

```
-- move the table to a new segment
-- indexes will be UNUSABLE and must be rebuilt
-- statistics must be regathered
ALTER TABLE hr.emp MOVE
 STORAGE ( INITIAL 20K
           NEXT 40K
           MINEXTENTS 2
           MAXEXTENTS 20
           PCTINCREASE 0 );
alter table emp MOVE TABLESPACE example ;
alter table emp move storage (
          NEXT 1024K
          PCTINCREASE 0 )
          COMPRESS;

-- allocating and deallocating extents
alter table hr.emp allocate extent;
alter table hr.emp deallocate extent;

-- allocating and deallocating extents
alter table hr.emp allocate extent;
alter table hr.emp deallocate unused;
```

```
 -- Marking Columns Unused
 ALTER TABLE hr.emp SET UNUSED (hiredate, mgr);
 SELECT * FROM DBA_UNUSED_COL_TABS;
 ALTER TABLE hr.emp DROP UNUSED COLUMNS statement;
 -- less undo space is consumed when checkpoint is done periodically
 ALTER TABLE hr.admin_emp DROP UNUSED COLUMNS CHECKPOINT 250;

-- read only and read write
ALTER TABLE employees READ ONLY;
ALTER TABLE employees READ WRITE;
select TABLE_NAME, READ_ONLY from user_tables where table_name='EMPLOYEES';
```

Performing Online Redefinition with DBMS_REDEFINITION

Redefining a Table

```
 /*
 This example illustrates online redefinition of the previously created table
 hr.admin_emp, which at this point only contains columns: empno, ename,
 job,deptno.
 The table is redefined as follows:
 -New columns mgr, hiredate, sal, and bonus are added.
 -The new column bonus is initialized to 0
 -The column deptno has its value increased by 10.
 -The redefined table is partitioned by range on empno.
 */
 --1 Verify that the table is a candidate for online redefinition.
 BEGIN
  DBMS_REDEFINITION.CAN_REDEF_TABLE(
  UNAME => 'HR',
  TNAME => 'ADMIN_EMP',
  OPTIONS_FLAG => dbms_redefinition.cons_use_pk); -- OR
 DBMS_REDEFINITION.CONS_USE_ROWID
 END;
 /
 -- 2. Create an interim table hr.int_admin_emp.
 CREATE TABLE hr.int_admin_emp
 (empno NUMBER(5) PRIMARY KEY,
  ename VARCHAR2(15) NOT NULL,
  job VARCHAR2(10),
  mgr NUMBER(5),
  hiredate DATE DEFAULT (sysdate),
  sal NUMBER(7,2),
  deptno NUMBER(3) NOT NULL,
  bonus NUMBER (7,2) DEFAULT(1000))
 PARTITION BY RANGE(empno)
 (PARTITION emp1000 VALUES LESS THAN (1000) TABLESPACE admin_tbs,
  PARTITION emp2000 VALUES LESS THAN (2000) TABLESPACE admin_tbs2);

 -- (Optional) If you are redefining a large table
 alter session force parallel dml parallel 2;
 alter session force parallel query parallel 2;

 --3. Start the redefinition process.
 BEGIN
```

```
 DBMS_REDEFINITION.START_REDEF_TABLE(
  UNAME => 'HR',
  ORIG_TABLE => 'ADMIN_EMP',
  INT_TABLE  => 'int_admin_emp',
  col_mapping=>'empno empno, ename ename, job job, deptno+10 deptno, 0
bonus', -- NULL if same columns in both tables
  options_flag => dbms_redefinition.cons_use_pk);
END;
/

-- 4 Copy dependent objects.
DECLARE
 N NUMBER;
BEGIN
DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS(
 uname =>'HR',
 orig_table =>'admin_emp',
 int_table => 'int_admin_emp',
 copy_indexes =>dbms_redefinition.cons_orig_params, -- 0 don't copy
 copy_triggers =>TRUE,
 copy_constraints =>TRUE,
 copy_privileges =>TRUE,
 ignore_errors => TRUE, -- default FALSE
 num_errors => n,
 copy_statistics => FALSE,  -- default FALSE
 copy_mvlog => FALSE);   -- default FALSE
-- bcuz ignore_errors=true
if n =0 then
 dbms_output.put_line('Success.');
else
 dbms_output.put_line('Number of Errors: ' || n);
end if;
END;
/

/* Note that the ignore_errors argument is set to TRUE for this call. The
reason is that the interim table was created with a primary key constraint,
and when COPY_TABLE_DEPENDENTS attempts to copy the primary key constraint
and index from the original table, errors occurs. You can ignore these
errors, but you must run the query shown in the next step to see if there are
other errors. */

-- 5. Query the DBA_REDEFINITION_ERRORS view to check for errors.
select object_name, base_table_name, ddl_txt from DBA_REDEFINITION_ERRORS;

-- 6. Optionally, synchronize the interim table hr.int_admin_emp
BEGIN
DBMS_REDEFINITION.SYNC_INTERIM_TABLE('hr', 'admin_emp', 'int_admin_emp');
END;
/

--7. Complete the redefinition.
BEGIN
 DBMS_REDEFINITION.FINISH_REDEF_TABLE('hr', 'admin_emp', 'int_admin_emp');
END;
/

/* The table hr.admin_emp is locked in the exclusive mode only for a small
```

```
 window toward the end of this step. After this call the table hr.admin_emp is
 redefined such that it has all the attributes of the hr.int_admin_emp table.
 */
 -- 8 Wait for any long-running queries against the interim table to complete,
 and then drop the interim table.
```

Redefining a Single Partition

```
/ *
We want to move the oldest partition of a range-partitioned sales table to a
tablespace named TBS_LOW_FREQ.
The table containing the partition to be redefined is defined as follows:
CREATE TABLE salestable
(s_productid NUMBER,
 s_saledate DATE,
 s_custid NUMBER,
 s_totalprice NUMBER)
TABLESPACE users
PARTITION BY RANGE(s_saledate)
(PARTITION sal03q1 VALUES LESS THAN (TO_DATE('01-APR-2003', 'DD-MON-YYYY')),
 PARTITION sal03q2 VALUES LESS THAN (TO_DATE('01-JUL-2003', 'DD-MON-YYYY')),
 PARTITION sal03q3 VALUES LESS THAN (TO_DATE('01-OCT-2003', 'DD-MON-YYYY')),
 PARTITION sal03q4 VALUES LESS THAN (TO_DATE('01-JAN-2004', 'DD-MON-YYYY')));
The table has a local partitioned index that is defined as follows:
CREATE INDEX sales_index ON salestable
(s_saledate, s_productid, s_custid) LOCAL;
*/

-- 1. Ensure that salestable is a candidate for redefinition.
BEGIN
DBMS_REDEFINITION.CAN_REDEF_TABLE(
 uname => 'STEVE',
 tname => 'SALESTABLE',
 options_flag => DBMS_REDEFINITION.CONS_USE_ROWID,
 part_name => 'sal03q1');
END;
/

-- 2 Create the interim table in the TBS_LOW_FREQ tablespace. Because this is
a redefinition of a range partition, the interim table is non-partitioned.
CREATE TABLE int_salestable
(s_productid NUMBER,
 s_saledate DATE,
 s_custid NUMBER,
 s_totalprice NUMBER)
 TABLESPACE tbs_low_freq;

-- 3. Start the redefinition process using rowid.
BEGIN
DBMS_REDEFINITION.START_REDEF_TABLE(
 uname => 'STEVE',
 orig_table => 'salestable',
 int_table => 'int_salestable',
 col_mapping => NULL,
 options_flag => DBMS_REDEFINITION.CONS_USE_ROWID,
 part_name => 'sal03q1');
END;
/
```

```
-- 4. Manually create any local indexes on the interim table.
CREATE INDEX int_sales_index ON int_salestable
(s_saledate, s_productid, s_custid)
 TABLESPACE tbs_low_freq;
-- 5. Optionally synchronize the interim table.
BEGIN
 DBMS_REDEFINITION.SYNC_INTERIM_TABLE(
 uname => 'STEVE',
 orig_table => 'salestable',
 int_table => 'int_salestable',
 part_name => 'sal03q1');
END;
/
-- 6. Complete the redefinition.
BEGIN
DBMS_REDEFINITION.FINISH_REDEF_TABLE(
 uname => 'STEVE',
 orig_table => 'salestable',
 int_table => 'int_salestable',
 part_name => 'sal03q1');
END;
/
-- 7. Wait for any long-running queries against the interim table to complete,
and then drop the interim table.
-- The following query shows that the oldest partition has been moved to the
new tablespace:
select partition_name, tablespace_name from user_tab_partitions where
table_name = 'SALESTABLE';
```

Migrating BasicFile LOBs to SecureFiles

```
 -- Grant privileges required for online redefinition.
 GRANT EXECUTE ON DBMS_REDEFINITION TO pm;
 GRANT ALTER ANY TABLE TO pm;
 GRANT DROP ANY TABLE TO pm;
 GRANT LOCK ANY TABLE TO pm;
 GRANT CREATE ANY TABLE TO pm;
 GRANT SELECT ANY TABLE TO pm;

 -- Privileges required to perform cloning of dependent objects.
 GRANT CREATE ANY TRIGGER TO pm;
 GRANT CREATE ANY INDEX TO pm;

 CONNECT pm

 CREATE TABLE cust(c_id NUMBER PRIMARY KEY,
                   c_zip NUMBER,
                   c_name VARCHAR(30) DEFAULT NULL,
                   c_lob CLOB);
 INSERT INTO cust VALUES(1, 94065, 'hhh', 'ttt');

 -- Creating Interim Table
 -- no need to specify constraints because they are copied over from the
 original table
 CREATE TABLE cust_int(c_id NUMBER NOT NULL,
                       c_zip NUMBER,
                       c_name VARCHAR(30) DEFAULT NULL,
                       c_lob CLOB)
```

```
                        LOB(c) STORE AS SECUREFILE (...);

declare
 col_mapping VARCHAR2(1000);
begin
 -- map all the columns in the interim table to the original table
 col_mapping :='c_id c_id , '||'c_zip c_zip , '||'c_name c_name, '||'c_lob
c_lob';
 DBMS_REDEFINITION.START_REDEF_TABLE('pm', 'cust', 'cust_int', col_mapping);
end;

declare
 error_count pls_integer := 0;
begin
 DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS('pm', 'cust', 'cust_int',
   1, TRUE,TRUE,TRUE,FALSE, error_count);
 DBMS_OUTPUT.PUT_LINE('errors := ' || TO_CHAR(error_count));
end;

exec DBMS_REDEFINITION.FINISH_REDEF_TABLE('pm', 'cust', 'cust_int');

-- Drop the interim table
DROP TABLE cust_int;
DESC cust;

-- to prove that the primary key on the c_id column is preserved after
migration.
INSERT INTO cust VALUES(1, 94065, 'hhh', 'ttt');
SELECT * FROM cust;
```

Using Flashback Drop and Managing the Recycle Bin

See Flashback Drop.

Managing Index-Organized Tables

```
-- Creating an Index-Organized Table
-- if the length of a row exceeds 20% of the index
--  block size, then the column that exceeded that threshold and
--  all columns after it are moved to the overflow segment.
--  Specify MAPPING TABLE to create a mapping table; it's needed to create a
bitmap index on the IOT
CREATE TABLE admin_docindex(
 token char(20),
 doc_id NUMBER,
 token_frequency NUMBER,
 token_offsets VARCHAR2(2000),
 CONSTRAINT pk_admin_docindex PRIMARY KEY (token, doc_id))
 ORGANIZATION INDEX TABLESPACE admin_tbs
  PCTTHRESHOLD 20
 OVERFLOW TABLESPACE admin_tbs2;
-- Creating Index-Organized Tables that Contain Object Types
CREATE OR REPLACE TYPE admin_typ AS OBJECT (col1 NUMBER, col2 VARCHAR2(6));
CREATE TABLE admin_iot (c1 NUMBER primary key, c2 admin_typ) ORGANIZATION
INDEX;
```

```
CREATE TABLE admin_iot2 OF admin_typ (col1 PRIMARY KEY) ORGANIZATION INDEX;
-- oken_offsets column value is always stored in the overflow area
TABLE admin_docindex2(
 token CHAR(20),
 doc_id NUMBER,
 token_frequency NUMBER,
 token_offsets VARCHAR2(2000),
 CONSTRAINT pk_admin_docindex2 PRIMARY KEY (token, doc_id))
 ORGANIZATION INDEX
 TABLESPACE admin_tbs
 PCTTHRESHOLD 20  -- default 50
 INCLUDING token_frequency
 OVERFLOW TABLESPACE admin_tbs2;

-- Parallelizing Index-Organized Table Creation
CREATE TABLE admin_iot3(i PRIMARY KEY, j, k, l)
 ORGANIZATION INDEX
 PARALLEL AS SELECT * FROM hr.jobs;

-- Altering Index-Organized Tables
ALTER TABLE admin_docindex INITRANS 4 OVERFLOW INITRANS 6;
ALTER TABLE admin_docindex PCTTHRESHOLD 15 INCLUDING doc_id;
-- if there's no existing overflow segment
ALTER TABLE admin_iot3 ADD OVERFLOW TABLESPACE admin_tbs2;

-- Moving (Rebuilding) Index-Organized Tables
ALTER TABLE admin_docindex MOVE; -- just rebuild
ALTER TABLE admin_docindex MOVE ONLINE;
ALTER TABLE admin_iot_lob MOVE LOB (admin_lob) STORE AS (TABLESPACE
admin_tbs3);
-- IOT is rebuilt along with its overflow segment
ALTER TABLE admin_docindex MOVE TABLESPACE admin_tbs2 OVERFLOW TABLESPACE
admin_tbs3;

-- Creating Secondary Indexes on Index-Organized Tables
CREATE INDEX Doc_id_index on Docindex(Doc_id, Token);
```

Managing External Tables

```
/* obtain info about external tables  */
select OWNER,TABLE_NAME,TYPE_OWNER,
 TYPE_NAME,DEFAULT_DIRECTORY_OWNER,DEFAULT_DIRECTORY_NAME,
 REJECT_LIMIT,ACCESS_TYPE,ACCESS_PARAMETERS, PROPERTY
from DBA_EXTERNAL_TABLES;
select OWNER, TABLE_NAME, LOCATION, DIRECTORY_OWNER, DIRECTORY_NAME
from DBA_EXTERNAL_LOCATIONS;


/* Creating an External Table and Loading Data */
CONNECT / AS SYSDBA;
-- Set up directories and grant access to hr
CREATE OR REPLACE DIRECTORY admin_dat_dir AS 'c:\temp\data';
CREATE OR REPLACE DIRECTORY admin_log_dir AS 'c:\temp\log';
CREATE OR REPLACE DIRECTORY admin_bad_dir AS 'c:\temp\bad';
GRANT READ ON DIRECTORY admin_dat_dir TO hr;
GRANT WRITE ON DIRECTORY admin_log_dir TO hr;
```

```
GRANT WRITE ON DIRECTORY admin_bad_dir TO hr;
-- hr connects. Provide the user password (hr) when prompted.
CONNECT hr
-- create the external table
CREATE TABLE admin_ext_employees
(employee_id NUMBER(4),
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 job_id VARCHAR2(10),
 manager_id NUMBER(4),
 hire_date DATE,
 salary NUMBER(8,2),
 commission_pct NUMBER(2,2),
 department_id NUMBER(4),
 email VARCHAR2(25))
ORGANIZATION EXTERNAL
(
 TYPE ORACLE_LOADER -- or ORACLE_DATAPUMP (for unload)
 DEFAULT DIRECTORY admin_dat_dir
ACCESS PARAMETERS
(
  records delimited by newline
  badfile admin_bad_dir:'empxt%a_%p.bad'
  logfile admin_log_dir:'empxt%a_%p.log'
  fields terminated by ','
  missing field values are null
( employee_id, first_name, last_name, job_id, manager_id,
  hire_date char date_format date mask "dd-mon-yyyy",
 salary, commission_pct, department_id, email )
)
LOCATION ('empxt1.dat', 'empxt2.dat')
)
PARALLEL -- useful to huge data
REJECT LIMIT UNLIMITED;

-- more options
-- in ACCESS PARAMETERS you can add
LOAD WHEN (job != MANAGER)

-- using SQL*Loader to generated script for creating external tables
-- (1) prepare control file such as:
LOAD DATA
INFILE *
INTO TABLE test_emp
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(employee_id,first_name,last_name,hire_date,salary,manager_id)
BEGINDATA
12345,"sam","alapati",sysdate,50000,99999
23456,"mark","potts",sysdate,50000,99999
-- (2) issue SQL*Loader using EXTERNAL_TABLE =GENERATE_ONLY NOT_USED EXECUTE
sqlldr USERID=system/sammyy1 CONTROL=test.ctl \
EXTERNAL_TABLE=GENERATE_ONLY
-- (3)  check the generated log file


/* Unloading into External Tables */
-- (using datapump drivers)
```

```
CREATE TABLE test_xt
 ORGANIZATION EXTERNAL(
 TYPE ORACLE_DATAPUMP
 DEFAULT DIRECTORY ext_data_dir
 ACCESS PARAMETERS (COMPRESSION ENABLED)
 LOCATION ('test_xt.dmp'))
AS
SELECT * FROM scott.dept;

-- encrypting loaded data
CREATE TABLE TEST
ORGANIZATION EXTERNAL (TYPE ORACLE_DATAPUMP DEFAULT DIRECTORY test_dir1
ACCESS PARAMETERS (ENCRYPTION ENABLED) LOCATION ('test.dmp'));

-- unload into multiple files
CREATE TABLE customers
ORGANIZATION EXTERNAL
(TYPE ORACLE_DATAPUMP DEFAULT DIRECTORY ext_data_dir ACCESS PARAMETERS
(NOBADFILE NOLOGFILE)
LOCATION ('customers1.exp', 'customers2.exp', 'customers3.exp',
'customers4.exp'))
PARALLEL 4 REJECT LIMIT UNLIMITED AS
SELECT c.*, co.country_name, co.country_subregion, co.country_region
FROM customers c, countries co where co.country_id=c.country_id;


-- enable parallel for loading (good if lots of data to load)
ALTER SESSION ENABLE PARALLEL DML;

-- load the data in hr employees table
INSERT INTO emp (employee_id, first_name, last_name, job_id, manager_id,
hire_date, salary, commission_pct, department_id, email)
SELECT * FROM admin_ext_employees;


/* Altering External Tables */
ALTER TABLE admin_ext_employees REJECT LIMIT 100;

-- access driver processes only the columns in the select list (default)
ALTER TABLE admin_ext_employees PROJECT COLUMN REFERNCED;
-- the access driver processes all of the columns
-- This setting always provides a consistent set of returned rows
ALTER TABLE admin_ext_employees PROJECT COLUMN ALL;

-- default directory
ALTER TABLE admin_ext_employees DEFAULT DIRECTORY admin_dat2_dir;

-- access parameter
ALTER TABLE admin_ext_employees ACCESS PARAMETERS (FIELDS TERMINATED BY ';');

-- location
ALTER TABLE admin_ext_employees LOCATION ('empxt3.txt', 'empxt4.txt');
```

Managing Indexes

Using Indexes

- When creating an extremely large index, consider allocating a larger temporary tablespace specially for the index creation.
- If you want the optimizer to consider the invisible indexes in its operation, you can set the new initialization parameter OPTIMIZER_USE_INVISIBLE_INDEXES to TRUE (the default is FALSE). You can set the parameter in the system and session levels.

```
-- obtain info about indexes
select * from DBA_INDEXES;
select * from DBA_IND_COLUMNS;
-- for funcation-based indexes
select * from DBA_IND_EXPRESSIONS;
-- optimizer stats
select * from DBA_IND_STATISTICS;
-- obtained from ANALYZE INDEX...VALIDATE STRUCTURE
select * from INDEX_STATS;

-- create index
CREATE INDEX emp_ename ON emp(ename)
TABLESPACE users
STORAGE (INITIAL 20K
 NEXT 20k
 PCTINCREASE 75);
CREATE INDEX emp_ename ON emp(ename) COMPUTE STATISTICS;
CREATE INDEX emp_name ON emp (mgr, emp1, emp2, emp3) ONLINE;
CREATE UNIQUE INDEX dept_unique_index ON dept (dname) TABLESPACE indx;
CREATE TABLE emp ( empno NUMBER(5) PRIMARY KEY, age INTEGER) ENABLE PRIMARY
KEY USING INDEX TABLESPACE users;
CREATE TABLE a ( a1 INT PRIMARY KEY USING INDEX (create index ai on a (a1)));
CREATE TABLE b( b1 INT, b2 INT,
 CONSTRAINT bu1 UNIQUE (b1, b2)
 USING INDEX (create unique index bi on b(b1, b2)),
 CONSTRAINT bu2 UNIQUE (b2, b1) USING INDEX bi);
CREATE TABLE c(c1 INT, c2 INT);
CREATE INDEX ci ON c (c1, c2);
ALTER TABLE c ADD CONSTRAINT cpk PRIMARY KEY (c1) USING INDEX ci;

-- bitmap index
Note: do not create it on a table that might be updated by more than one
session, otherwise you risk seeing ORA-02049 error now and then!
CREATE BITMAP INDEX gender_idx ON employee(gender) TABLESPACE emp_index_05;

-- Reverse-Key Indexes
-- good when you do sequential insertion of values into the index
CREATE INDEX reverse_idx ON employee(emp_id) REVERSE;

-- function based
CREATE INDEX area_index ON rivers (area(geo));

-- compress
CREATE INDEX emp_ename ON emp(ename) TABLESPACE users COMPRESS 1;
ALTER INDEX emp_ename REBUILD NOCOMPRESS;
```

```
-- Invisible Index (11g)
CREATE INDEX name_indx ON employees(emp_name) INVISIBLE;
SELECT /*+ index (EMP_NAME NAME_INDEX) */ ...
ALTER INDEX name_indx VISIBLE;
ALTER INDEX name_indx INVISIBLE;
SELECT INDEX_NAME, VISIBILITY FROM DBA_INDEXES WHERE INDEX_NAME='NAME_INDX';

-- Bitmap Join Index (see next section).

-- rebuilding an index
ALTER INDEX emp_name REBUILD;
ALTER INDEX emp_name REBUILD ONLINE;

-- Monitoring Index Usage
-- as the index owner
-- Note: generally, do not drop an index build on a FK column
ALTER INDEX index MONITORING USAGE;
ALTER INDEX index NOMONITORING USAGE;
Select INDEX_NAME,TABLE_NAME,MONITORING,USED,START_MONITORING,END_MONITORING
From V$OBJECT_USAGE -- as the index owner
WHERE INDEX_NAME='EMP_LNAME_IND';

-- Monitoring Space Use of Indexes
-- 1. Analyzing statistics
exec DBMS_STATS.GATHER_INDEX_STATS( OWNNAME=>'HR', INDNAME=>'EMP_LNAME_IND')
-- 2. Validating the index
ANALYZE INDEX EMP_LNAME_IND VALIDATE STRUCTURE;
-- 3. Checking PCT_USED
SELECT PCT_USED FROM INDEX_STATS WHERE NAME = 'EMP_LNAME_IND';
-- 4. Dropping and rebuilding (or coalescing) the index
ALTER INDEX emp_name REBUILD;

-- estimating index space
-- gather index and its underlying table statistics
SET SERVEROUTPUT ON
declare
 l_index_ddl VARCHAR2(1000);
 l_used_bytes NUMBER;
 l_allocated_bytes NUMBER;
BEGIN
 DBMS_SPACE.create_index_cost (
  DDL => 'create index persons_idx on persons(person_id)',
  USED_BYTES => l_used_bytes,
  ALLOC_BYTES => l_allocated_bytes);
 DBMS_OUTPUT.PUT_LINE ('used = ' || to_char(l_used_bytes/1024) || ' KB ' || '
allocated = ' || to_char(l_allocated_bytes/1024) || ' KB');
END;
/
```

Using Bitmap Join Indexes (BJI)

- A bitmap join index is a bitmap index for the join of two or more tables. It is designed when joining fact tables with dimension tables as in warehouse databases (start schema model).

```
CREATE BITMAP INDEX NAMES_CTR_bji
  ON NAMES2(COUNTRIES.COUNTRY_NAME)
```

```
  FROM NAMES2, COUNTRIES
  WHERE NAMES2.NAT_ID=COUNTRIES.ID
  NOLOGGING COMPUTE STATISTICS;

SELECT /*+ INDEX_COMBINE(N NAMES_CTR_bji)*/
 C.COUNTRY_NAME
 FROM NAMES2 N, COUNTRIES C
 WHERE N.NAT_ID = C.ID;
```

Partitioned Indexes

```
 -- Global Indexes
 -- can be nonpartitioned or partitioned indexes
 CREATE INDEX ticketsales_idx ON
 ticket_sales(month) GLOBAL
 PARTITION BY range(month)
 (PARTITION ticketsales1_idx VALUES LESS THAN (3)
  PARTITION ticketsales1_idx VALUES LESS THAN (6)
  PARTITION ticketsales2_idx VALUES LESS THAN (9)
  PARTITION ticketsales3_idx VALUES LESS THAN (MAXVALUE));

 CREATE INDEX hgidx
 ON tab (c1,c2,c3) GLOBAL
 PARTITION BY HASH (c1,c2)
 (PARTITION p1 TABLESPACE ts1,
  PARTITION p2 TABLESPACE ts2,
  PARTITION p3 TABLESPACE ts3,
  PARTITION p4 TABLESPACE ts4);

 -- Local Indexes
 CREATE INDEX ticket_no_idx ON
 ticket_sales(ticket__no) LOCAL TABLESPACE localidx_01;
```

Managing Materialized Views

Obtaining Information about Materialized Views

```
-- all info about the mviews
SELECT
 OWNER, MVIEW_NAME, CONTAINER_NAME, QUERY, QUERY_LEN,
 UPDATABLE, UPDATE_LOG, MASTER_ROLLBACK_SEG, MASTER_LINK, REWRITE_ENABLED,
 REWRITE_CAPABILITY, REFRESH_MODE, REFRESH_METHOD, BUILD_MODE,
FAST_REFRESHABLE,
 LAST_REFRESH_TYPE, LAST_REFRESH_DATE, STALENESS, AFTER_FAST_REFRESH,
UNKNOWN_PREBUILT,
 UNKNOWN_PLSQL_FUNC, UNKNOWN_EXTERNAL_TABLE, UNKNOWN_CONSIDER_FRESH,
UNKNOWN_IMPORT,
 UNKNOWN_TRUSTED_FD, COMPILE_STATE, USE_NO_INDEX, STALE_SINCE
FROM DBA_MVIEWS
ORDER BY OWNER, MVIEW_NAME


-- list mview last refreshed on year or longer ago
--  Note: they cause the mview logs linked to them to get so large by time
SELECT  OWNER, MVIEW_NAME,
 TO_CHAR(LAST_REFRESH_DATE,'DD-MM-YY') LAST_REFRESHED_DATE
FROM DBA_MVIEWS
WHERE LAST_REFRESH_DATE < SYSDATE-365
ORDER BY LAST_REFRESH_DATE ASC


-- determine the master table (and last refresh time)
select OWNER, NAME, MASTER_OWNER, MASTER, LAST_REFRESH
 from DBA_SNAPSHOT_REFRESH_TIMES


-- FAST_REFRESHABLE mviews which couldn't make FAST refresh
--  in their last refreshes
SELECT
 OWNER, MVIEW_NAME, REFRESH_METHOD, FAST_REFRESHABLE,
 LAST_REFRESH_TYPE, LAST_REFRESH_DATE, STALENESS, AFTER_FAST_REFRESH,
 COMPILE_STATE, STALE_SINCE
FROM DBA_MVIEWS
WHERE FAST_REFRESHABLE<>'NO' and LAST_REFRESH_TYPE<>'FAST'
ORDER BY OWNER, MVIEW_NAME


-- all MATERIALIZED VIEW LOGS
SELECT
 LOG_OWNER, MASTER, LOG_TABLE, LOG_TRIGGER, ROWIDS,
 PRIMARY_KEY, OBJECT_ID, FILTER_COLUMNS,
 SEQUENCE, INCLUDE_NEW_VALUES
FROM DBA_MVIEW_LOGS
ORDER BY  LOG_OWNER, MASTER
```

Monitoring the Progress of a Materialized View Refresh

- Reference: Document ID 258021.1

```
/* Determine if a Specific MVIEW is Being Refreshed */
```

```
select o.owner, o.object_name mview, username, s.sid
from v$lock l, dba_objects o, v$session s
where o.object_id=l.id1 and
      l.type='JI' and
      l.lmode=6   and
      s.sid=l.sid and
      o.object_type='TABLE';



/* Determine if a Refresh Group is Being Refreshed */
-- Given the name of the refresh group and its owner, the
--  following query can be used to identify if a refresh
--  is being executed by a job queue process:
   select s.sid, s.username
   from dba_jobs_running jr, v$session s, dba_jobs j
   where jr.sid=s.sid and
         j.job=jr.job and
         upper(j.what) like '%REFRESH%<name of the refresh group>%';

-- to determine if a refresh has been executed manually (
--  not through a background job )
  select username, sid, rowner, rname
   from ( select username, s.sid, rc.rowner, rc.rname, count(*)
          from v$lock l, dba_objects o, v$session s,
               dba_refresh_children rc
          where o.object_id=l.id1 and
                l.type='JI' and
                l.lmode=6 and
                s.sid=l.sid and
                o.object_type='TABLE' and
                o.object_name=rc.name and
                o.owner=rc.owner and
                rc.type='SNAPSHOT'
          group by username, s.sid, rc.rowner, rc.rname
          having count(*)=( select count(*) from dba_refresh_children
                            where rowner= rc.rowner and rname=rc.rname and
                                  type='SNAPSHOT') );


/* Determine which MVIEW in a Refresh Group is Being Refreshed */
select SID, SERIAL#, CURRMVOWNER, CURRMVNAME
  from v$mvrefresh



/* Determine the Current Phase of a Refresh */
-- if in Propagation phase (push and purge)
--  locks are released once push and purge finished
-- in Push phase:
   select l.sid,
          decode( count(*), 0, 'No propagation in progress',
                               'Propagation in progress' ) State
   from v$lock l, dbms_lock_allocated la
   where l.type='UL' and
         l.lmode=4 and
         l.id1=la.lockid and
```

```
         la.name='ORA$DEF$EXE$PushCommonLock'
   group by l.sid;

-- identifying the target site of the propagation:
   select l.sid, 'Currently propagating to ' || substr(la.name, 13)
   from v$lock l, dbms_lock_allocated la
   where l.type='UL' and
         l.lmode=6 and
         l.id1=la.lockid and
         la.name like 'ORA$DEF$EXE$%';

-- in Purge phase:
--  if refresh is invoked with purge_option>0 and parallelism>0
select l.sid,
          decode( count(*), 0, 'No purge in progress',
                               'Purge is in progress' ) State
   from v$lock l, dbms_lock_allocated la
   where l.type='UL' and
         l.lmode=6 and
         l.id1=la.lockid and
         la.name='ORA$DEF$EXE$PurgeCommonLock'
   group by l.sid;


-- Refresh Subphases ( SETUP, INSTATNIATION and WRAPUP)
-- to get:
          - the type of the ongoing refresh
          - the phase of the ongoing refresh
          - the number of DMLs performed by the refresh
select  CURRMVOWNER_KNSTMVR || '.' || CURRMVNAME_KNSTMVR
              "MVIEW BEING REFRESHED",
           decode( REFTYPE_KNSTMVR, 1, 'FAST', 2, 'COMPLETE', 'UNKNOWN' )
REFTYPE,
           decode(GROUPSTATE_KNSTMVR, 1, 'SETUP', 2, 'INSTANTIATE',
                                      3, 'WRAPUP', 'UNKNOWN' ) STATE,
           TOTAL_INSERTS_KNSTMVR INSERTS,
           TOTAL_UPDATES_KNSTMVR UPDATES,
           TOTAL_DELETES_KNSTMVR DELETES
   from X$KNSTMVR X
   WHERE type_knst=6 and
         exists (select 1 from v$session s
                 where s.sid=x.sid_knst and
                       s.serial#=x.serial_knst);


/* Steps to Determine Whether a Refresh is Hanging, or Moving Slowly */
-- if the refresh in Propagation:
-- check for blocking sessions:
-- to identify if a row being pushed is locked by a users session:
select
 s.username,
 w.holding_session holder,
 h.username holder_name,
 s.row_wait_row# rowno,
 o.object_name,
 o.owner
from dba_waiters w, dba_objects o, v$session s, v$session h
```

```
where w.waiting_session = s.sid
and w.holding_session = h.sid
and s.row_wait_obj# = o.object_id;

-- Check the wait events
-- if refresh is being executed from DBMS_JOB:
select sw.sid, sw.event, sw.p1, sw.p2
from v$session_wait sw
where sw.sid in ( select qs.sid
from dba_jobs_running jr, dba_jobs j,
v$px_session qs
where jr.job = j.job
and jr.sid = qs.qcsid
and upper(j.what) like '%..%')
and sw.wait_time = 0;

-- if refresh is being executed by a users session:
select sw.sid, sw.event, sw.p1, sw.p2
from v$session_wait sw, v$session s
where sw.sid = s.sid
and sw.sid in (select qs.sid from v$px_session qs)
and sw.wait_time = 0


-- Check if a large error is being queued:
-- to identify if a large error is being queued at the remote master site:
select count(*)
from v$sqlarea a, v$session s
where s.sql_address = a.address
and s.sql_hash_value = a.hash_value
and a.sql_text like '%DEF$_AQERROR%'
and s.username = 'REPADMIN';


-- if refresh in WRAPUP phase:
--  get the wait event :
--  in normal cases, it's read scattered or sequential:
   select event, p1, p2, p3
   from v$session_wait
   where sid=<SID of the session at master site>;

-- the object that is being accessed by:
select owner, segment_name
   from dba_extents
   where file_id=<P1> and
         <P2> between block_id and block_id+blocks-1;

-- if the accessed object is MView log, check the following section
--   which diagnoses the MView Log


-- if refresh in INSTANTIATION phase:
-- the rows are pulled from the master site
-- check the wait:
select event, p1, p2, p3
   from v$session_wait
   where sid=<SID of the session at master site>;
```

```
-- if the event is enqueue: check the blocking session
select
 s.username,
 w.holding_session holder,
 h.username holder_name,
 s.row_wait_row# rowno,
 o.object_name,
 o.owner
from dba_waiters w, dba_objects o, v$session s, v$session h
where w.waiting_session = s.sid
and w.holding_session = h.sid
and s.row_wait_obj# = o.object_id;

-- if event is SQL*Net message from dblink:
the session at master site should be checked for the type of the wait
if the wait is "SQL*Net message from dblink" in both sites,
submit a SR in Oracle Support.
```

Materialized View Typical Refresh Errors

```
ORA-12004: "REFRESH FAST cannot be used for ...
This error indicates a problem with the log at master. See Note:179469.1 for
further information.

ORA-12034: "snapshot log on "%s"."%s" younger than last refresh"
This error also indicates a problem with the log at master. See Note:204127.1
for further information.

ORA-23402: refresh was aborted because of conflicts caused by deferred txns
This error is caused by outstanding conflicts logged in the DefError table at
the master. This can be workaround by setting refresh_after_errors to true.
See Note:1031119.6 and Note:39232.1 for the details.

ORA-23385: replication parallel push heap_size argument not valid This error
is caused if the heap_size value is set to NULL.  Query rgroup$ to obtain the
current value of heap_size and use dbms_refresh.change to set  it to not null
value. IE. 0.  See Note:49558.1 for the error definition.
```

Using Materialized Views

- Good reference is Note ID 179466.1
- If USER1 wants to refresh MView created by USER2 and MView Master Table is owned by USER3, then the following privileges are required:

```
grant SELECT ANY TABLE to USER2;
grant CREATE TABLE, CREATE ANY MATERIALIZED VIEW to USER2;
grant FLASHBACK on USER3.OAS_DOCLINE to USER1;
grant FLASHBACK on USER3.OAS_DOCLINE to USER2;
grant FLASHBACK on USER3.MLOG$_OAS_DOCLINE to USER1;
grant FLASHBACK on USER3..MLOG$_OAS_DOCLINE to USER2;
```

For COMPLETE refresh, to avoid logging, you can consider setting ATOMIC_REFRESH=false. It makes the refresh faster.

When creating materialized view, you should keep in mind the following restrictions:

- The defining query of the materialized view cannot contain any non-repeatable expressions (ROWNUM, SYSDATE, non-repeatable PL/SQL functions, and so on).

- The query cannot contain any references to RAW or LONG RAW datatypes or object REFs

Restrictions on Fast Refresh

- The defining query of the materialized view is restricted as follows:
- The materialized view must not contain references to non-repeating expressions like SYSDATE and ROWNUM.
- The materialized view must not contain references to RAW or LONG RAW data types.
- It cannot contain a SELECT list subquery.
- It cannot contain analytical functions (for example, RANK) in the SELECT clause.
- It cannot contain a MODEL clause.
- It cannot contain a HAVING clause with a subquery.
- It cannot contain nested queries that have ANY, ALL, or NOT EXISTS.
- It cannot contain a [START WITH …] CONNECT BY clause.
- It cannot contain multiple detail tables at different sites.
- ON COMMIT materialized views cannot have remote detail tables.
- Nested materialized views must have a join or aggregate.

If you received the following error when trying to refresh a materialized view, trying compiling it:

```
ORA-00942: table or view does not exist
ORA-06512: at "SYS.DBMS_SNAPSHOT",
alter materialized view … compile
```

```
-- required privs
GRANT CREATE MATERIALIZED VIEW TO hr;
GRANT QUERY REWRITE TO hr;

-- mv log
-- you must specify the ROWID, SEQUENCE and the INCLUDING NEW VALUES
--  For aggregate materialized views, it must also contain every column
--   in the table referenced in the materialized view
CREATE MATERIALIZED VIEW LOG
ON employees WITH SEQUENCE, ROWID
(employee_id,department_id,salary) INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG
ON departments WITH SEQUENCE, ROWID
(department_id, department_name) INCLUDING NEW VALUES;

ALTER MATERIALIZED VIEW LOG ON sales ADD ROWID
(prod_id, cust_id, time_id, amount_sold) INCLUDING NEW VALUES;

DROP MATERIALIZED VIEW LOG ON sales;

-- create mv
-- the options
BUILD IMMEDIATE | DEFERRED
REFRESH NEVER, FAST, FORCE, COMPLETE - on commit or on demand
ENABLE QUERY REWRITE -- is a must for rewriting queries. DISABLE
WITH PRIMARY KEY
START WITH ROUND(SYSDATE + 1) + 11/24  NEXT NEXT_DAY(TRUNC(SYSDATE), 'MONDAY')
+ 15/24
recommended to gather stats on it then
EXECUTE DBMS_STATS.GATHER_TABLE_STATS ( 'HR','DEPT01_MV', estimate_percent
```

```
=> 20, block_sample   => TRUE, cascade   => TRUE);

-- (1) Materialized Views with Aggregates
--  vm log must also contain every column
--   in the table referenced in the materialized view
--  there must be a COUNT(*) and a COUNT(column) on any aggregated columns
CREATE MATERIALIZED VIEW dept01_mv
(department_id, department_name, salary_count, average_salary)
 TABLESPACE ts1
BUILD IMMEDIATE
REFRESH FAST
ENABLE QUERY REWRITE
AS
select d.department_id, d.department_name,
 count(e.salary), -- it is a must for mv with aggregate or  count(*),
 avg(e.salary)
from departments d, employees e
where d.department_id = e.department_id
group by d.department_id, d.department_name;

CREATE MATERIALIZED VIEW product_sales_mv
 TABLESPACE ts1
BUILD IMMEDIATE -- DEFERRED
REFRESH FAST -- FAST, FORCE, COMPLETE - on commit or on demand
ENABLE QUERY REWRITE
AS
SELECT p.prod_name, SUM(s.amount_sold) AS dollar_sales,
COUNT(*) AS cnt, COUNT(s.amount_sold) AS cnt_amt
FROM sales s, products p
WHERE s.prod_id = p.prod_id GROUP BY p.prod_name;


-- (2) Materialized Views Containing Only Joins
-- contain only joins and no aggregates
-- ROWID column must be present in each materialized view log and mv SELECT
CREATE MATERIALIZED VIEW LOG ON sales WITH ROWID;
CREATE MATERIALIZED VIEW LOG ON times WITH ROWID;
CREATE MATERIALIZED VIEW LOG ON customers WITH ROWID;
CREATE MATERIALIZED VIEW detail_sales_mv
PARALLEL
BUILD IMMEDIATE
REFRESH FAST AS
SELECT s.rowid "sales_rid", t.rowid "times_rid", c.rowid "customers_rid",
       c.cust_id, c.cust_last_name, s.amount_sold, s.quantity_sold, s.time_id
FROM sales s, times t, customers c
WHERE  s.cust_id = c.cust_id(+) AND s.time_id = t.time_id(+);


-- (3) Nested Materialized Views
-- a materialized view whose definition is based on another materialized view
CREATE MATERIALIZED VIEW LOG ON sales WITH ROWID;
CREATE MATERIALIZED VIEW LOG ON customers WITH ROWID;
CREATE MATERIALIZED VIEW LOG ON times WITH ROWID;
/*create materialized view join_sales_cust_time as fast refreshable at
   COMMIT time */
CREATE MATERIALIZED VIEW join_sales_cust_time
REFRESH FAST ON COMMIT AS
```

```
SELECT c.cust_id, c.cust_last_name, s.amount_sold, t.time_id,
       t.day_number_in_week, s.rowid srid, t.rowid trid, c.rowid crid
FROM sales s, customers c, times t
WHERE s.time_id = t.time_id AND s.cust_id = c.cust_id;
/* create materialized view log on join_sales_cust_time
    and include the required columns */
CREATE MATERIALIZED VIEW LOG ON join_sales_cust_time
WITH ROWID (cust_last_name, day_number_in_week, amount_sold)
INCLUDING NEW VALUES;
/* create the single-table aggregate materialized view sum_sales_cust_time
on join_sales_cust_time as fast refreshable at COMMIT time */
CREATE MATERIALIZED VIEW sum_sales_cust_time
REFRESH FAST ON COMMIT AS
SELECT COUNT(*) cnt_all, SUM(amount_sold) sum_sales, COUNT(amount_sold)
       cnt_sales, cust_last_name, day_number_in_week
FROM join_sales_cust_time
GROUP BY cust_last_name, day_number_in_week;
-- Refreshing a Nested Materialized View
DBMS_MVIEW.REFRESH('SALES_MV,COST_MV', nested => TRUE);

/* commenting on mvs */
COMMENT ON MATERIALIZED VIEW sales_mv IS 'sales materialized view';
SELECT MVIEW_NAME, COMMENTS
FROM USER_MVIEW_COMMENTS WHERE MVIEW_NAME = 'SALES_MV';
```

Using Query Rewriting

- A query is rewritten only when a certain number of conditions are met:
  - Query rewrite must be enabled for the session.
  - A materialized view must be enabled for query rewrite.
  - The rewrite integrity level should allow the use of the materialized view.
- Use Column Alias Lists in the CREATE MATERIALIZED VIEW command to let the query rewrite work

```
/* Required Privs */
grant query rewrite to hr;
-- if the mv uses a table from other schema
grant query rewrite on employees to scott;
or
grant global query rewrite to hr;

/* Ensuring that Query Rewrite Takes Effect */
-- (1) query rewrite enabled for the mv
select REWRITE_ENABLED  from user_mviews where mview_name='DEPT01_MV';
ALTER MATERIALIZED VIEW dept01_mv ENABLE QUERY REWRITE ;
-- (2) QUERY_REWRITE_ENABLED = TRUE | FORCE
ALTER SESSION SET QUERY_REWRITE_ENABLED=true;
select get_par('query_rewrite_enabled') from dual;
-- for user with dba priv
select value from v$parameter where upper(name)='QUERY_REWRITE_ENABLED';
alter system set QUERY_REWRITE_ENABLED=TRUE;
-- (3) OPTIMIZER_MODE = ALL_ROWS, FIRST_ROWS, or FIRST_ROWS_n, (where n = 1,
10, 100, 1000)
select get_par('OPTIMIZER_MODE') from dual;
-- (4) OPTIMIZER_FEATURES_ENABLE must be 10.0.0 or higher
select value from v$parameter where upper(name)='OPTIMIZER_FEATURES_ENABLE';
```

```
/* Column Alias */
-- failed example
CREATE MATERIALIZED VIEW sales_mv
ENABLE QUERY REWRITE AS
SELECT s.time_id sales_tid, c.time_id costs_tid
FROM sales s, products p, costs c
WHERE s.prod_id = p.prod_id AND c.prod_id = p.prod_id AND
      p.prod_name IN (SELECT prod_name FROM products);
-- the right way
CREATE MATERIALIZED VIEW sales_mv (sales_tid, costs_tid)
ENABLE QUERY REWRITE AS
SELECT s.time_id, c.time_id
FROM sales s, products p, costs c
WHERE s.prod_id = p.prod_id AND c.prod_id = p.prod_id AND
      p.prod_name IN (SELECT prod_name FROM products);

/* Rewrite Integrity */
-- QUERY_REWRITE_INTEGRITY: ENFORCED (returned data is 100% correct), TRUSTED,
STALE_TOLERATED

/* Verifying that Query Rewrite has Occurred */
EXPLAIN PLAN FOR
select d.department_id, d.department_name,
 count(e.salary),
 avg(e.salary)
from departments d, employees e
where d.department_id = e.department_id
group by d.department_id, d.department_name;

-- if rewriting occurred, you should see operation: MAT_VIEW REWRITE ACCESS
SELECT OPERATION, OBJECT_NAME FROM PLAN_TABLE;

/* Using the EXPLAIN_REWRITE Procedure with Query Rewrite */
-- target: to know why rewriting didn't occur
-- 1) create EXPLAIN_REWRITE table
@/oracle/oradb10g/rdbms/admin/utlxrw.sql
-- 2) execute
declare
 v varchar2(4000);
begin
v := 'select d.department_id, d.department_name, count(e.salary),
avg(e.salary) from departments d, employees e where d.department_id =
e.department_id group by d.department_id, d.department_name';
DBMS_MVIEW.EXPLAIN_REWRITE(
 query => v,
 mv =>'', -- you can specify the comma-separated mv(s) or NULL to consider all
mvs
 statement_id => 'ID6');  -- client-supplied unique identifier to distinguish
output messages
end;
/

SELECT message, original_cost, rewritten_cost
 FROM rewrite_table ORDER BY sequence;

-- alternatively to see neat output use SYS.XRW (see Wareshouse documentation)
```

ReWrite Hints

```
/* REWRITE and NOREWRITE Hints */
SELECT /*+ NOREWRITE */ ...
SELECT /*+ REWRITE (sum_sales_pscat_week_mv) */  ...

/* The Rewrite_or_Error Hint */
-- use the mv or error
SELECT /*+ REWRITE_OR_ERROR */
 prod_id, SUM(quantity_sold) AS sum_sales_qty
FROM sales_data
GROUP BY prod_id;

/* NO_MULTIMV_REWRITE hint prevents the query from being rewritten with more
than one materialized
   NO_BASETABLE_MULTIMV_REWRITE hint prevents the query from being rewritten
with a combination of materialized views and the base tables
 */
```

Using EXPLAIN_MVIEW Procedure: Viewing Materialized View Capabilities

- Target: to determine if a materialized view is fast refreshable and what types of query rewrite you can perform with a particular materialized view.

```

DBMS_MVIEW.EXPLAIN_MVIEW (
 mv -- mv, a SELECT or a CREATE MATERIALIZED VIEW statement
statement_id -- any id

--(1) create MV_CAPABILITIES_TABLE
@/oracle/oradb10g/rdbms/admin/utlxmv.sql
--(2) execute
declare
 v varchar2(4000);
begin
 v := 'DEPT01_MV';
 DBMS_MVIEW.EXPLAIN_MVIEW( V,'ID01');
 Commit;
end;
/

SELECT CAPABILITY_NAME, POSSIBLE, RELATED_TEXT,RELATED_NUM, MSGNO, MSGTXT
FROM MV_CAPABILITIES_TABLE
WHERE STATEMENT_ID='ID01'
ORDER BY SEQ
/

SELECT MSGTXT
FROM MV_CAPABILITIES_TABLE
WHERE STATEMENT_ID='ID01'
ORDER BY SEQ
/
```

Using DBMS_ADVISOR.TUNE_MVIEW

- The TUNE_MVIEW procedure shows how to decompose a materialized view into two or more materialized views or to restate the materialized view in a way that is more advantageous for fast refresh and query rewrite. TUNE_MVIEW analyzes and processes the input statement and generates two sets of output results: one for the materialized view implementation and the other for undoing the CREATE materialized view operations.

- ADVISOR privilege required to execute the procedure.

```
declare
 task_cust_mv VARCHAR2(30);
 create_mv_ddl VARCHAR2(4000);
begin
 task_cust_mv := 'cust_mv';
 create_mv_ddl := 'CREATE MATERIALIZED VIEW REPTEST.MP_OAS_BALANCE_MV
BUILD IMMEDIATE
REFRESH FORCE ON DEMAND
WITH PRIMARY KEY
as
SELECT cmpcode     || LPAD (TO_CHAR (yr), 4, 0)           "BAL_KEY",
       CMPCODE,
       YR,
       PERIOD,
       BALCODE,
       CURCODE,
       REPBASIS,
       CURFLAG,
       ACCODE,
       DEBIT_VALUE,
       CREDIT_VALUE,
       1 UNMARKER,
       ROWID RID
  FROM OAS_BALANCE
  WHERE CMPCODE NOT IN (''XYZ'', ''KH'', ''GUI'')';
 DBMS_ADVISOR.TUNE_MVIEW(task_cust_mv, create_mv_ddl);
end;
/

-- Access IMPLEMENTATION Output
SELECT * FROM USER_TUNE_MVIEW WHERE TASK_NAME= 'cust_mv' AND
SCRIPT_TYPE='IMPLEMENTATION';

-- Save IMPLEMENTATION Output in a Script File
CREATE DIRECTORY TUNE_RESULTS AS '/myscript'
GRANT READ, WRITE ON DIRECTORY TUNE_RESULTS TO PUBLIC;
EXECUTE DBMS_ADVISOR.CREATE_FILE(DBMS_ADVISOR.GET_TASK_SCRIPT('cust_mv'),
'TUNE_RESULTS', 'mv_create.sql');
```

Registering a User-defined Table as Materialized View

```
-- table and mv have the same name
CREATE TABLE sum_sales_tab
PCTFREE 0  TABLESPACE demo
AS
SELECT s.prod_id, SUM(amount_sold) AS dollar_sales, SUM(quantity_sold) AS
unit_sales FROM sales s GROUP BY s.prod_id;

CREATE MATERIALIZED VIEW sum_sales_tab_mv
ON PREBUILT TABLE WITHOUT REDUCED PRECISION
ENABLE QUERY REWRITE AS
SELECT s.prod_id, SUM(amount_sold) AS dollar_sales,
    SUM(quantity_sold) AS unit_sales
FROM sales s GROUP BY s.prod_id;
```

Managing Clusters and Hash Clusters

- Hashing is useful when you have the following conditions:
  - Most queries are equality queries on the cluster key:

    SELECT … WHERE cluster_key = …;
  - The tables in the hash cluster are primarily static in size so that you can determine the number of rows and amount of space required for the tables in the cluster.

```
-- obtaining info on clusters
select * from DBA_CLUSTERS;
-- map table columns to cluster columns
select * from DBA_CLU_COLUMNS;

-- create cluster
CREATE CLUSTER emp_dept (deptno NUMBER(3))
 SIZE 600
 TABLESPACE users
 STORAGE (INITIAL 200K
 NEXT 300K
 MINEXTENTS 2
 MAXEXTENTS 20
 PCTINCREASE 33);
CREATE TABLE emp (
empno NUMBER(5) PRIMARY KEY,
ename VARCHAR2(15) NOT NULL,
. . .
deptno NUMBER(3) REFERENCES dept)
CLUSTER emp_dept (deptno);
CREATE TABLE dept (
deptno NUMBER(3) PRIMARY KEY, . . . )
CLUSTER emp_dept (deptno);

-- create cluster index
CREATE INDEX emp_dept_index
 ON CLUSTER emp_dept
 TABLESPACE users
 STORAGE (INITIAL 50K
 NEXT 50K
 MINEXTENTS 2
 MAXEXTENTS 10
 PCTINCREASE 33);

-- dropping a cluster
-- no tables
DROP CLUSTER emp_dept;
-- there are tables
DROP CLUSTER emp_dept INCLUDING TABLES;
DROP CLUSTER emp_dept INCLUDING TABLES CASCADE CONSTRAINTS;
-- drop a table in the cluster
drop table dept;

/* Managing Hash Clusters */

-- obtain info
selet * from DBA_CLUSTERS ;
```

```
selet * from DBA_CLU_COLUMNS;
selet * from DBA_CLUSTER_HASH_EXPRESSIONS;

-- Creating Hash Clusters
CREATE CLUSTER trial_cluster (trialno NUMBER(5,0))
 TABLESPACE users
 STORAGE (INITIAL 250K NEXT 50K
 MINEXTENTS 1 MAXEXTENTS 3
 PCTINCREASE 0)
 HASH IS trialno HASHKEYS 150;
CREATE TABLE trial (
trialno NUMBER(5,0) PRIMARY KEY,
...)
CLUSTER trial_cluster (trialno);

-- Creating a Sorted Hash Cluster
CREATE CLUSTER call_detail_cluster (
 telephone_number NUMBER,
 call_timestamp NUMBER SORT,
 call_duration NUMBER SORT )
 HASHKEYS 10000 HASH IS telephone_number
 SIZE 256;
CREATE TABLE call_detail (
 telephone_number NUMBER,
 call_timestamp NUMBER SORT,
 call_duration NUMBER SORT,
 other_info VARCHAR2(30) )
CLUSTER call_detail_cluster (
 telephone_number, call_timestamp, call_duration );
-- following select gets advantage from the sorted hash cluster
SELECT * WHERE telephone_number = 6505551212;

-- Creating Single-Table Hash Clusters
CREATE CLUSTER peanut (variety NUMBER)
SIZE 512 SINGLE TABLE HASHKEYS 500;

-- Dropping Hash Clusters
DROP CLUSTER emp_dept;
```

Managing Views, Sequences, and Synonyms

```
-- create views
CREATE VIEW sales_staff AS
 SELECT empno, ename, deptno FROM emp WHERE deptno = 10
WITH CHECK OPTION CONSTRAINT sales_staff_cnst;
-- Creating Views with Errors
CREATE FORCE VIEW AS ...;

-- creating sequence
CREATE SEQUENCE emp_sequence
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 NOCYCLE
 CACHE 10;
-- alter a sequence
ALTER SEQUENCE emp_sequence
INCREMENT BY 10
MAXVALUE 10000
CYCLE
CACHE 20;

-- Creating Synonyms
CREATE PUBLIC SYNONYM public_emp FOR jward.emp;
```

Managing Transactions

Implementing Oracle's Concurrency Control

Oracle Isolaction Levels

- There are four possible isolation levels:
  - Serializable: doesn't allow any concurrent.
  - Repeatable read
  - Read uncommitted
  - Read committed (Default): Oracle only guarantees statement-level isolation here (changes between reads are seen), not transaction-level isolation.

command	Its effect
`SET TRANSACTION READ ONLY`	When set, the transaction that follows operates on essentially a snapshot of the database at the time the command was issued. This is especially useful when multiple select statements are executed over the course of a transaction, and data must be consistent.
`SET TRANSACTION READ WRITE`	Set the transactgion back to read write
`SET TRANSACTION ISOLATION LEVEL READ COMMITTED`	if we try to modify a record that already has a DML Row Exclusive lock on that record, the attempt to update will wait until the locks are released.
`SET TRANSACTION ISOLATION LEVEL SERIALIZABLE`	if we try to modify a record that already has a DML Row Exclusive lock on that record, the attempt to update will fail

```
-- must be the first command in an transaction
-- to enable transaction-level isolation
ALTER SESSION SET ISOLATION LEVEL SERIALIZABLE;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
-- alternatively ( repeatable reads )
SET TRANSACTION READ ONLY
```

Oracle Lock Types

- DML Locks: Row-Level Lock: Exclusive Table-Level Lock: Row exclusive
- DDL Locks: DDL is not allowed if DML Lock was there and "ORA-00054: resource bus" will be returned unless DDL_LOCK_TIMEOUT is defined in seconds (11g).
- Explicit Table Locking: DDL requires Table exclusive DML lock. You can specify that a DDL command wait for a specific length of time before it fails:

  `LOCK TABLE ... IN lockmode MODE [NOWAIT | WAIT integer]`

  lockmode: `[ROW] SHARE, [[SHARE] ROW] EXCLUSIVE, SHARE UPDATE`

  integer in seconds
- Latches: are internal mechanisms that protect shared data structures in the SGA.
- Data dictionary locks: whenever the dictionary objects are being modified.

- Distributed locks: used in a distributed database system or in RAC.
- Internal locks: are used by Oracle to protect access to structures such as datafiles, tablespaces, and rollback segments.

```
-- current locks in the DB
select SID,
DECODE(TO_CHAR(BLOCK),'0','Not-Blocking','1','Blocking') IS_BLOCKING,
DECODE(TYPE,'TM','DML enqueue','TX','Transaction enqueue','UL','User
supplied',TYPE) LOCK_TYPE,
DECODE(TO_CHAR(LMODE),'0','None','1','Null','2','Row-S (SS)','3','Row-X
(SX)','4','Share (S)','5','S/Row-X (SSX)','6','Exclusive (X)') HELD_LMODE,
DECODE(TO_CHAR(REQUEST),'0','None','1','Null','2','Row-S (SS)','3','Row-X
(SX)','4','Share (S)','5','S/Row-X (SSX)','6','Exclusive (X)') REQUEST_LMODE
from v$lock;

-- Oracle 11g: to allow DDL wait for lock instead or returning error
ALTER SESSION SET ddl_lock_timeout = 30;

-- explicit table locking (to acquire an exclusive lock=no updates)
-- to release the lock: ROLLBACK, COMMIT
lock table emp in EXCLUSIVE mode nowait ;
```

Identifying Blocking Sessions

```
-- oracle supplied script printing blocking sessions in tree-like view
@$ORACLE_HOME/rdbms/admin/utllockt.sql

select SID,
DECODE(TYPE,'TM','DML enqueue','TX','Transaction enqueue','UL','User
supplied',TYPE) LOCK_TYPE,
DECODE(TO_CHAR(LMODE),'0','None','1','Null','2','Row-S (SS)','3','Row-X
(SX)','4','Share (S)','5','S/Row-X (SSX)','6','Exclusive (X)') HELD_LMODE,
DECODE(TO_CHAR(REQUEST),'0','None','1','Null','2','Row-S (SS)','3','Row-X
(SX)','4','Share (S)','5','S/Row-X (SSX)','6','Exclusive (X)') REQUEST_LMODE
from v$lock
where BLOCK=1;

SELECT sid, username, blocking_session blocking_sid
FROM V$SESSION WHERE blocking_session_status='VALID';
```

Using Autonomous Transaction

- The autonomous transactions give you the ability to commit or roll back the subprogram's changes independent of the main program.

```
-- write errors log
CREATE OR REPLACE PROCEDURE error_log
 (error__msg in varchar2, procedure_name IN VARCHAR2) IS
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
 INSERT INTO log_table (error_msg, procedure_name)
 VALUES (error_msg,procedure_name));
 COMMIT;
EXCEPTION
WHEN OTHERS THEN
 ROLLBACK;
END error_log;
/
```

```
-- using DDL statements in triggers
-- performing an audit of database queries
-- performing an audit of and failed (unauthorized) database activity
CREATE OR REPLACE TRIGGER aud_bef_trig
 BEFORE INSERT ON emp FOR EACH ROW
DECLARE
 PRAGMA AUTONOMOUS_TRANSACTION
BEGIN
 INSERT INTO audit_employee VALUES (
  :new.username, 'before insert', sysdate);
 COMMIT;
END;
/
```

Managing Long Transactions with Workspace Manager

- Workspace Manager is by default installed when you create a database using DBCA but not manually. To check, make sure WMSYS exists in DBA_USERS.
- With Workspace Manager, you can maintain multiple versions of data, which you can keep or discard as necessary.
- Its benifits:
  - To enable simultaneous read and write access to data during long transactions.
  - To create multiple data scenarios for what-if analyses.
- The original table will be renamed to tableName_LT, new table tableName_AUX will be created. Users be default are placed in LIVE workspace.
- **Refresh**: update the data in the refreshed workspace with the changes made in its parent workspace.
- **Merge**: update the parent workspace with the changes made to the merged workspace.
- Reference: Application Developer's Guide - Workspace Manager

```
-- should return LIVE
select dbms_wm.getworkspace from dual;

/* go back in time within a workspace */
-- the example is using the LIVE workspace

-- privs
begin
  dbms_wm.grantSystemPriv (
    'ACCESS_ANY_WORKSPACE,    ' ||
    'MERGE_ANY_WORKSPACE,     ' ||
    'CREATE_ANY_WORKSPACE,    ' ||
    'REMOVE_ANY_WORKSPACE,    ' ||
    'ROLLBACK_ANY_WORKSPACE'  ,
    'USER1'                 ,
    'YES');
end;
/


conn user1
```

```
create table dept
( DEPTNO NUMBER(3) PRIMARY KEY,
  DNAME  VARCHAR2(14) )
/

create table EMP
( EMPNO NUMBER(4) PRIMARY KEY, -- PK is mandatory
  ENAME VARCHAR2(10),
  DEPTNO NUMBER(3) references dept(deptno))
/


-- set version enabled tables
-- VIEW_WO_OVERWRITE = without Overwrite
begin
  dbms_wm.enableVersioning('EMP','VIEW_WO_OVERWRITE');
  dbms_wm.enableVersioning('DEPT','VIEW_WO_OVERWRITE');
end;
/

select TABLE_NAME, STATE, HISTORY
from USER_WM_VERSIONED_TABLES
order by TABLE_NAME ;


-- testing data
-- Note: rows cannot be inserted if the Workspace isn't
--  in its latest version by DBMS_WM.GotoSavePoint();
insert into dept values (1,'d1');
commit;
insert into emp values (1,'e1',1);
insert into emp values (2,'e2',1);
commit;

select * from emp, dept where emp.deptno=dept.deptno;

-- save current time
var dt_1 varchar2(21)

begin
  select to_char(sysdate, 'dd.mm.yyyy hh24:mi:ss') into :dt_1 from dual;
end;
/

-- insert extra rows
insert into dept values(2,'d2');
commit;
insert into emp values(3,'e3',2);
insert into emp values(4,'e4',2);
commit;

select * from emp, dept where emp.deptno=dept.deptno;

-- return to dt_1 time
begin
  dbms_wm.gotoDate(to_date(:dt_1, 'dd.mm.yyyy hh24:mi:ss'));
end;
```

```
/
commit;

select * from emp, dept where emp.deptno=dept.deptno;

-- disable versioning
begin
  dbms_wm.disableVersioning('DEPT');
  dbms_wm.disableVersioning('EMP');
end;
/
```

```
/* Creating Workspaces and Merging/Refreshing Data */
-- privs
begin
  dbms_wm.grantSystemPriv (
    'ACCESS_ANY_WORKSPACE,    ' ||
    'MERGE_ANY_WORKSPACE,     ' ||
    'CREATE_ANY_WORKSPACE,    ' ||
    'REMOVE_ANY_WORKSPACE,    ' ||
    'ROLLBACK_ANY_WORKSPACE'  ,
    'USER1'                 ,
    'YES');
end;
/


conn user1

create table dept
( DEPTNO NUMBER(3) PRIMARY KEY,
  DNAME  VARCHAR2(14) )
/

create table EMP
( EMPNO NUMBER(4) PRIMARY KEY, -- PK is mandatory
  ENAME VARCHAR2(10),
  DEPTNO NUMBER(3) references dept(deptno))
/

insert into dept values (1,'d1');
commit;
insert into emp values (1,'e1',1);
insert into emp values (2,'e2',1);
commit;

-- set version enabled tables
-- VIEW_WO_OVERWRITE = without Overwrite
-- this option make WAREHOUSES_HIST view contain
-- complete history information about data changes
begin
  -- both table must be version enabled because they are linked
  dbms_wm.enableVersioning('DEPT,EMP','VIEW_WO_OVERWRITE');
end;
/

-- create workspaces
```

```
begin
  dbms_wm.createworkspace ('WS1');
  dbms_wm.createworkspace ('WS2');
end;
/

select workspace, parent_workspace from user_workspaces;

-- move logged on user to WS1
begin
  dbms_wm.gotoworkspace('WS1');
end;
/

select * from emp, dept where emp.deptno=dept.deptno;

-- insert extra rows
insert into dept values(2,'d2');
commit;
insert into emp values(3,'e3',2);
insert into emp values(4,'e4',2);
commit;

select * from emp, dept where emp.deptno=dept.deptno;

-- go back to Live workspace
begin
  dbms_wm.gotoworkspace('LIVE');
end;
/

select * from emp, dept where emp.deptno=dept.deptno;

-- change some data within the LIVE workspaces
insert into dept values(3,'d3');
commit;
insert into emp values(5,'e5',3);
insert into emp values(6,'e6',3);
commit;


select * from emp, dept where emp.deptno=dept.deptno;

-- move to Workspace WS2
begin
  dbms_wm.gotoworkspace('WS2');
end;
/

-- you won't see changes made in the LIVE workspace nor the changes
--  made in the WS1 workspace.
select * from emp, dept where emp.deptno=dept.deptno;

insert into dept values(4,'d3');
commit;
insert into emp values(7,'e7',4);
insert into emp values(8,'e8',4);
```

```
commit;

-- REFRESH WS1
begin
 dbms_wm.refreshworkspace('WS1');
end;
/

-- move to WS1
begin
  dbms_wm.gotoworkspace('WS1');
end;
/

select * from emp, dept where emp.deptno=dept.deptno;

-- MERGE WS2
begin
  dbms_wm.mergeworkspace('WS2');
end;
/

-- verify merge succeeded
begin
  dbms_wm.gotoworkspace('LIVE');
end;
/

select * from emp, dept where emp.deptno=dept.deptno;


-- cleaning up
begin
  dbms_wm.disableversioning ('dept, emp', true);
end;
/

begin
  dbms_wm.gotoworkspace  ('LIVE');
  dbms_wm.removeworkspace('WS1');
  dbms_wm.removeworkspace('WS2');
end;
/
```

```
/* Resolving Conflicts */
-- privs
begin
  dbms_wm.grantSystemPriv (
    'ACCESS_ANY_WORKSPACE,    ' ||
    'MERGE_ANY_WORKSPACE,     ' ||
    'CREATE_ANY_WORKSPACE,    ' ||
    'REMOVE_ANY_WORKSPACE,    ' ||
    'ROLLBACK_ANY_WORKSPACE'  ,
    'USER1'                 ,
    'YES');
end;
/
```

```
conn user1

create table dept
( DEPTNO NUMBER(3) PRIMARY KEY,
  DNAME  VARCHAR2(14) )
/

create table EMP
( EMPNO NUMBER(4) PRIMARY KEY, -- PK is mandatory
  ENAME VARCHAR2(10),
  DEPTNO NUMBER(3) references dept(deptno))
/

insert into dept values (1,'d1');
insert into dept values (2,'d2');
commit;
insert into emp values (1,'e1',1);
insert into emp values (2,'e2',1);
insert into emp values (3,'e3',2);
insert into emp values (4,'e4',2);
commit;

-- set version enabled tables
begin
  dbms_wm.enableVersioning('DEPT,EMP','VIEW_WO_OVERWRITE');
end;
/


-- create workspace and goto it
begin
  dbms_wm.createworkspace('WS1');
end;
/
begin
  dbms_wm.gotoworkspace('WS1');
end;
/

-- make an update
update emp set ename='in WS1' where empno=1;
commit;

-- go back to LIVE workspace
begin
  dbms_wm.gotoworkspace('LIVE');
end;
/

-- make an update
update emp set ename='in LIVE' where empno=1;
commit;

-- to see the conflicts in xxx_CONF, you should leave LIVE workspace
begin
  dbms_wm.gotoworkspace('WS1');
```

```
end;
/

column wm_workspace format a16
select * from EMP_CONF;


/* To resolve the conflict */
-- (1) call dbms_wm.beginresolve
begin
  dbms_wm.BeginResolve('WS1');
end;
/

-- (2) call dbms_wm.resolveconflicts
begin
  -- Keep update from Parent (LIVE)
  dbms_wm.resolveconflicts(
    workspace    => 'WS1', -- workspace to check for conflicts with others
    table_name   => 'EMP',
    where_clause => 'empno=1',
    keep         => 'PARENT');
end;
/
-- OR
begin
  dbms_wm.resolveconflicts(
    workspace    => 'WS1',
    table_name   => 'EMP',
    where_clause => 'empno=1',
    keep         => 'CHILD');
end;
/
commit;

-- (3) call dbms_wm.commitresolve.
begin
  dbms_wm.CommitResolve('WS1');
end;
/


-- let's see if we still have a conflict:
select * from emp_conf;

-- check the data
select * from emp ;

-- let's see what data in LIVE
begin
  dbms_wm.gotoworkspace('LIVE');
end;
/

select * from emp ;

-- if data modified in WS1 was chosen, merge it with LIVE
```

```
begin
  dbms_wm.mergeworkspace('WS1');
end;
/

-- Cleaning up
begin
  dbms_wm.gotoworkspace('LIVE');
  dbms_wm.removeworkspace('WS1');
  dbms_wm.disableversioning('emp,dept');
end;
/
```

Repairing Corrupted Data

Options for Repairing Data Block Corruption

- Drop and re-create an object after the corruption is detected.
- If data block corruption is limited to a subset of rows, then another option is to rebuild the table by selecting all data except for the corrupt rows.
- Use the DBMS_REPAIR package to detect and repair corrupt blocks in tables and indexes.
- Use Flashback Versions Query to query values of a row over a period of time. Then arrange undo plan.
- Block Media Recovery BMR

Detecting Corruptions Methods

- DB_VERIFY utility
- ANALYZE TABLE .. VALIDATE STRUCTURE [CASCADE]
- Setting the initialization parameters: DB_BLOCK_CHECKING and DB_BLOCK_CHECKSUM
- exp utility
- DBMS_REPAIR
- Data Recovery Advisor automatically runs after any corruption is detected and can alos proactively invoked. See Using Data Recovery Advisor in RMAN.

Using dbv (DBVerify) Utility

- The utility checks only for logical corruption below the HWM.

```
dbv file=D:\ORACLE\ORADATA\ORA10G\USERS01.DBF  blocksize=4096
dbv file=example01.dbf blocksize=8192
..
examine "Total Pages Marked Corrupt"
```

Setting the Initialization Parameters for Detecting Corruption

Verifying Block Integrity in Real Time: DB_BLOCK_CHECKING

- Block checking typically causes 1% to 10% overhead, depending on workload.
- You should set DB_BLOCK_CHECKING to FULL if the performance overhead is acceptable.
- Possible Values:
  - OFF: No block checking (except for SYSTEM) (default)
  - LOW: Basic block header checks are performed after block contents change in memory
  - MEDIUM: All LOW checks, as well as block checking for all non-index-organized table blocks, are performed.
  - FULL: All LOW and MEDIUM checks, as well as checks on index blocks, are performed.

```
alter system set DB_BLOCK_CHECKING=FULL;
```

```
alter session set DB_BLOCK_CHECKING=LOW;
```

Verifying Block Integrity in Real Time: DB_BLOCK_CHECKSUM

- Can prevent corruption caused by underlying I/O systems
- FULL setting causes 4-5% overhead

```
DB_BLOCK_CHECKSUM = { OFF | FALSE | TYPICAL | TRUE |FULL }
select value from v$parameter where upper(name)='DB_BLOCK_CHECKSUM';
```

Detecting lost write: DB_LOST_WRITE_PROTECT

- A data block lost write occurs when an I/O subsystem acknowledges the completion of the block write, while in fact the write did not occur in the persistent storage.

```
DB_LOST_WRITE_PROTECT = { NONE | TYPICAL | FULL }
```

Settubg the DB_ULTRA_SAFE Parameter (In Oracle 11g)

This parameter is used to set the effective values of the parameters: `DB_BLOCK_CHECKING`, `DB_LOST_WRITE_PROTECT`, `DB_BLOCK_CHECKSUM`. This parameter takes one of the following values:

`off` — this value means any values you set for any of the three parameters will not be overridden.

`data only` — The effective value of the parameters will be as follows:

Parameter	Active Value
DB_BLOCK_CHECKING	medium
DB_LOST_WRITE_PROTECT	typical
DB_BLOCK_CHECKSUM.	full

`data and index` — The effective value of the parameters will be as follows:

Parameter	Active Value
DB_BLOCK_CHECKING	full
DB_LOST_WRITE_PROTECT	typical
DB_BLOCK_CHECKSUM.	full

Using ANALYZE Command

- Validates the structure of a table or table partitions, and index or index partitions.
- Does not mark blocks as soft corrupt; only reports them
- When corruption detected, an error message returned.

```
ANALYZE TABLE table_name VALIDATE STRUCTURE CASCADE;
```

Using EXP to Detect Corruption

- exp utility reports an errors when it encounters corruption.

```
$ exp hr/hr tables=departments
About to export specified tables via Conventional Path ...
. . .
EXP-00056: ORACLE error 1578 encountered
ORA-01578: ORACLE data block corrupted (file # 5, block # 51)
```

```
ORA-01110: data file 5: '/u01/app/oracle/oradata/..'
```

Using DBMS_REPAIR

DBMS_REPAIR Limitations and Restrictions

- DBMS_REPAIR procedures have the following limitations:
  - Tables with LOB datatypes, nested tables, and varrays are supported, but the out of line columns are ignored.
  - Clusters are supported in the SKIP_CORRUPT_BLOCKS and REBUILD_FREELISTS procedures, but not in the CHECK_OBJECT procedure.
  - Index-organized tables and LOB indexes are not supported.
  - The DUMP_ORPHAN_KEYS procedure does not operate on bitmap indexes or function-based indexes.
  - The DUMP_ORPHAN_KEYS procedure processes keys that are no more than 3,950bytes long.

Evaluate the Costs and Benefits of Using DBMS_REPAIR

Before using DBMS_REPAIR you must weigh the benefits of its use in relation to the liabilities. You should also examine other options available for addressing corrupt objects. Begin by answering the following questions:

- What is the extent of the corruption?

To determine if there are corruptions and repair actions, execute the CHECK_OBJECT procedure and query the repair table.

- What other options are available for addressing block corruptions? Consider the following:
  - If the data is available from another source, then drop, re-create, and repopulate the object.
  - Issue the CREATE TABLE...AS SELECT statement from the corrupt table to create a new one.
  - Ignore the corruption by excluding corrupt rows from SELECT statements.
  - Perform media recovery.
- What logical corruptions or side effects are introduced when you use DBMS_REPAIR to make an object usable? Can these be addressed? What is the effort required to do so?

It is possible that you do not have access to rows in blocks marked corrupt. However, a block can be marked corrupt even if there are rows that you can validly access.

It is also possible that referential integrity constraints are broken when blocks are marked corrupt. If this occurs, then disable and reenable the constraint; any inconsistencies are reported. After fixing all problems, you should be able to reenable the constraint.

Logical corruption can occur when there are triggers defined on the table. For example, if rows are reinserted, should insert triggers be fired or not? You can address these issues only if you understand triggers and their use in your installation.

If indexes and tables are not synchronized, then execute the DUMP_ORPHAN_KEYS procedure to obtain information from the keys that might be useful in rebuilding corrupted data. Then issue the ALTER INDEX...REBUILD ONLINE statement to synchronize the table with its indexes.

- If repair involves loss of data, can this data be retrieved?

You can retrieve data from the index when a data block is marked corrupt. The DUMP_ORPHAN_KEYS procedure can help you retrieve this information.

Detect and Report Corruptions using DBMS_REPAIR

```
 -- Reports corruptions and identifies the associated repair directives
 -- 1. build the repair table
 BEGIN
  DBMS_REPAIR.ADMIN_TABLES (
    table_name => 'REPAIR_TABLE',
    table_type => DBMS_REPAIR.REPAIR_TABLE,
    action => DBMS_REPAIR.CREATE_ACTION,
    tablespace => 'USERS');
 END;
 /

SELECT OBJECT_NAME, BLOCK_ID, CORRUPT_TYPE, MARKED_CORRUPT,
CORRUPT_DESCRIPTION, REPAIR_DESCRIPTION FROM REPAIR_TABLE;

 -- 2. report corruptions
 SET SERVEROUTPUT ON
 DECLARE
  num_corrupt INT;
 BEGIN
  num_corrupt := 0;
  DBMS_REPAIR.CHECK_OBJECT (
    schema_name => 'HR',
    object_name => 'DEPARTMENTS',
    repair_table_name => 'REPAIR_TABLE',
    corrupt_count => num_corrupt);
 END;
 /

 -- 3. make the object usable: ignoring corruptions during table and index
 scans
 -- mark corrupted blocks
 SET SERVEROUTPUT ON
  DECLARE num_fix INT;
 BEGIN
  num_fix := 0;
  DBMS_REPAIR.FIX_CORRUPT_BLOCKS (
    schema_name => 'HR',
    object_name => 'DEPARTMENTS',
    object_type => DBMS_REPAIR.TABLE_OBJECT,
    repair_table_name => 'REPAIR_TABLE',
    fix_count => num_fix);
 END;
 /

SELECT OBJECT_NAME, BLOCK_ID, MARKED_CORRUPT FROM REPAIR_TABLE;

 -- skip blocks marked as corrupted
 BEGIN
  DBMS_REPAIR.SKIP_CORRUPT_BLOCKS (
   schema_name => 'HR',
   object_name => 'DEPARTMENTS',
   object_type  => DBMS_REPAIR.TABLE_OBJECT,
   flags => DBMS_REPAIR.SKIP_FLAG); -- or NOSKIP_FLAG
 END;
 /
```

```
select SKIP_CORRUPT from DBA_TABLES
 where table_name='DEPARTMENTS' and woner='HR';

-- 4. reports on index entries that point to rows in corrupt data blocks
--  may be useful for rebuilding lost rows
BEGIN
 DBMS_REPAIR.ADMIN_TABLES (
   table_name => 'ORPHAN_KEY_TABLE',
   table_type => DBMS_REPAIR.ORPHAN_TABLE,
   action => DBMS_REPAIR.CREATE_ACTION,
   tablespace => 'USERS');
END;
/

SET SERVEROUTPUT ON
DECLARE num_orphans INT;
BEGIN
 num_orphans := 0;
 DBMS_REPAIR.DUMP_ORPHAN_KEYS(
  schema_name => 'SCOTT',
  object_name => 'PK_DEPT',
  object_type => DBMS_REPAIR.INDEX_OBJECT,
  repair_table_name => 'REPAIR_TABLE',
  orphan_table_name => 'ORPHAN_KEY_TABLE',
  key_count => num_orphans);
 DBMS_OUTPUT.PUT_LINE('Orphan Key Count: ' ||TO_CHAR(num_orphans));
END;
/
```

Managing Automated Database Maintenance Tasks

Predefined Automated Maintenance Tasks

- Automatic Optimizer Statistics Collection
- Automatic Segment Advisor
- Automatic SQL Tuning Advisor

Predefined Maintenance 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.

Obtaining Information about Predefined Maintenance Tasks

```
-- currently running Scheduler jobs created for automated maintenance tasks
select
CLIENT_NAME,JOB_NAME,JOB_SCHEDULER_STATUS,TASK_NAME,TASK_TARGET_TYPE,TASK_TARG
ET_NAME,TASK_PRIORITY,TASK_OPERATION
From DBA_AUTOTASK_CLIENT_JOB ;

-- automated maintenance task statistics over 7-day and 30-day periods
select
CLIENT_NAME,STATUS,CONSUMER_GROUP,CLIENT_TAG,PRIORITY_OVERRIDE,ATTRIBUTES,WIND
OW_GROUP,SERVICE_NAME,RESOURCE_PERCENTAGE,USE_RESOURCE_ESTIMATES,MEAN_JOB_DURA
TION,MEAN_JOB_CPU,MEAN_JOB_ATTEMPTS,MEAN_INCOMING_TASKS_7_DAYS,MEAN_INCOMING_T
ASKS_30_DAYS,TOTAL_CPU_LAST_7_DAYS,TOTAL_CPU_LAST_30_DAYS,MAX_DURATION_LAST_7_
DAYS,MAX_DURATION_LAST_30_DAYS,WINDOW_DURATION_LAST_7_DAYS,WINDOW_DURATION_LAS
T_30_DAYS
From DBA_AUTOTASK_CLIENT

-- history of automated maintenance task job runs
select
CLIENT_NAME,WINDOW_NAME,WINDOW_START_TIME,WINDOW_DURATION,JOB_NAME,JOB_STATUS,
JOB_START_TIME,JOB_DURATION,JOB_ERROR,JOB_INFO
from DBA_AUTOTASK_JOB_HISTORY
order by WINDOW_START_TIME desc;

-- Lists the windows that belong to MAINTENANCE_WINDOW_GROUP
select
WINDOW_NAME,WINDOW_NEXT_TIME,WINDOW_ACTIVE,AUTOTASK_STATUS,OPTIMIZER_STATS,SEG
MENT_ADVISOR,SQL_TUNE_ADVISOR,HEALTH_MONITOR
from DBA_AUTOTASK_WINDOW_CLIENTS;
```

```
-- per-window history of job execution counts for each
--  automated maintenance task
select
CLIENT_NAME,WINDOW_NAME,WINDOW_START_TIME,WINDOW_DURATION,JOBS_CREATED,JOBS_ST
ARTED,JOBS_COMPLETED,WINDOW_END_TIME
from DBA_AUTOTASK_CLIENT_HISTORY;
```

Enabling and Disabling Maintenance Tasks

```
 select CLIENT_NAME,STATUS,CONSUMER_GROUP,CLIENT_TAG,PRIORITY_OVERRIDE,
 ATTRIBUTES,WINDOW_GROUP,SERVICE_NAME,RESOURCE_PERCENTAGE,
 USE_RESOURCE_ESTIMATES,MEAN_JOB_DURATION,MEAN_JOB_CPU,MEAN_JOB_ATTEMPTS,
 MEAN_INCOMING_TASKS_7_DAYS,MEAN_INCOMING_TASKS_30_DAYS,TOTAL_CPU_LAST_7_DAYS,
 TOTAL_CPU_LAST_30_DAYS,MAX_DURATION_LAST_7_DAYS,MAX_DURATION_LAST_30_DAYS,
 WINDOW_DURATION_LAST_7_DAYS,WINDOW_DURATION_LAST_30_DAYS
 from DBA_AUTOTASK_CLIENT;

 -- disable
 BEGIN
 DBMS_AUTO_TASK_ADMIN.DISABLE(
  client_name => 'sql tuning advisor',
  operation => NULL,
  window_name => NULL); -- null means all widows
 END;
 /

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

 -- enable or disable ALL automated maintenance tasks for ALL windows
 EXECUTE DBMS_AUTO_TASK_ADMIN.DISABLE;
 EXECUTE DBMS_AUTO_TASK_ADMIN.ENABLE;

 -- Enabling and Disabling Maintenance Tasks for Specific Maintenance Windows
 BEGIN
 DBMS_AUTO_TASK_ADMIN.DISABLE(
  client_name => 'sql tuning advisor',
  operation => NULL,
  window_name => 'MONDAY_WINDOW');
 END;
 /
```

Configuring Maintenance Windows

```
 -- Modifying a Maintenance Window
 --  see Using Windows SECTION
 BEGIN
 dbms_scheduler.disable( name => 'SATURDAY_WINDOW');
```

```
dbms_scheduler.set_attribute(
 name => 'SATURDAY_WINDOW',
 attribute => 'DURATION',
 value => numtodsinterval(4, 'hour'));
 dbms_scheduler.enable(
 name => 'SATURDAY_WINDOW');
END;
/

-- Creating a New Maintenance Window
BEGIN
DBMS_SCHEDULER.CREATE_WINDOW(
 window_name => 'EARLY_MORNING_WINDOW',
 duration => numtodsinterval(1, 'hour'),
 resource_plan => 'DEFAULT_MAINTENANCE_PLAN',
 repeat_interval => 'FREQ=DAILY;BYHOUR=5;BYMINUTE=0;BYSECOND=0');
dbms_scheduler.add_window_group_member(
 group_name => 'MAINTENANCE_WINDOW_GROUP',
 window_list => 'EARLY_MORNING_WINDOW');
END;
/


-- Removing a Maintenance Window
-- window continues to exist but no longer runs automated maintenance tasks.
--  Any other Scheduler jobs assigned to this window continue to run as
usual.
BEGIN
DBMS_SCHEDULER.REMOVE_WINDOW_GROUP_MEMBER(
 group_name => 'MAINTENANCE_WINDOW_GROUP',
 window_list => 'EARLY_MORNING_WINDOW');
END;
/
```

Managing Resources

- The steps:
    1. Create a pending area.
    2. Create a resource consumer group.
    3. Create a resource plan.
    4. Create a plan directive.
    5. Validate the pending area.
    6. Submit the pending area

```
-- required privs
exec DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SYSTEM_PRIVILEGE -
(GRANTEE_NAME => 'scott', PRIVILEGE_NAME => 'ADMINISTER_RESOURCE_MANAGER');

-- creating demo users
conn system
create user u1 identified by u1 default tablespace users;
create user u2 identified by u2 default tablespace users;
create user u3 identified by u3 default tablespace users;
grant resource, connect to u1;
grant resource, connect to u2;
grant resource, connect to u3;

CREATE TABLE U1.EMP AS SELECT * FROM HR.EMPLOYEES;
CREATE TABLE U2.EMP AS SELECT * FROM HR.EMPLOYEES;
CREATE TABLE U3.EMP AS SELECT * FROM HR.EMPLOYEES;


-- Creating a Pending Area
exec dbms_resource_manager.create_pending_area();

-- if you make mistakes
exec dbms_resource_manager.clear_pending_area;

-- Resource Consumer Groups
-- pre-configured CGs:
--     DEFAULT_CONSUMER_GROUP,OTHER_GROUPS,SYS_GROUP,LOW_GROUP

begin
 dbms_resource_manager.create_consumer_group (
       CONSUMER_GROUP => 'PROGRAMMERS',
       COMMENT => 'IT programmers',
       CPU_MTH => 'ROUND-ROBIN'  -- other possible value is RUN_TO_COMPLETION
                       );
end;
/

begin
 dbms_resource_manager.create_consumer_group (
                       CONSUMER_GROUP => 'CLERKS',
                       COMMENT => 'Group of Clerks',
                       CPU_MTH => 'ROUND-ROBIN'
                       );
```

```
end;
/


set linesize 100
column consumer_group format a15
column comments format a40
column cpu_method format a11
select DISTINCT consumer_group, cpu_method, comments
from dba_rsrc_consumer_groups
order by 1
/

-- Consumer Group Mapping
-- Assigning User Sessions to Consumer Groups
begin
 dbms_resource_manager.set_consumer_group_mapping(
       ATTRIBUTE => dbms_resource_manager.oracle_user, -- it is constant (no
quots)
       VALUE => 'U1',
       CONSUMER_GROUP => 'PROGRAMMERS');
end;
/

begin
 dbms_resource_manager.set_consumer_group_mapping(
       ATTRIBUTE => dbms_resource_manager.oracle_user, -- it is constant (no
quots)
       VALUE => 'U2',
       CONSUMER_GROUP => 'CLERKS');
end;
/

-- possible attributes are:
CLIENT_OS_USER
CLIENT_PROGRAM
CLIENT_MACHINE
MODULE_NAME
MODULE_NAME_ACTION
ORACLE_USER
SERVICE_NAME
SERVICE_MODULE

-- /* for demo only
begin
 dbms_resource_manager.set_consumer_group_mapping(
       ATTRIBUTE => CLIENT_OS_USER,
       VALUE => 'graciej',
       CONSUMER_GROUP => 'OLAP_GROUP');
end;
/
*/


-- Establishing Mapping Priorities
begin
dbms_resource_manager.SET_CONSUMER_GROUP_MAPPING_PRI(
```

```
 EXPLICIT => 1,
 CLIENT_OS_USER => 5,
 CLIENT_MACHINE => 3,
 CLIENT_PROGRAM => 4,
 ORACLE_USER => 2,
 MODULE_NAME => 6,
 MODULE_NAME_ACTION => 7,
 SERVICE_NAME => 8,
 SERVICE_MODULE => 9,
 SERVICE_MODULE_ACTION => 10);
end;
/
-- ERROR
BEGIN
DBMS_RESOURCE_MANAGER.SET_MAPPING_PRIORITY(
 EXPLICIT => 1, CLIENT_MACHINE => 2, MODULE_NAME => 3, ORACLE_USER => 4,
SERVICE_NAME => 5, CLIENT_OS_USER => 6, CLIENT_PROGRAM => 7,
MODULE_NAME_ACTION => 8, SERVICE_MODULE=>9, SERVICE_MODULE_ACTION=>10);
END;
/


-- Resource Plans
-- Simple Resource Plan (single-level resource plans for allocating CPU)
/* 10g (deprecated in 11g) */
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_SIMPLE_PLAN(
 SIMPLE_PLAN => 'JOBS_PLAN',
 CONSUMER_GROUP1 => 'PROGRAMMERS',
 GROUP1_CPU => 75,
 CONSUMER_GROUP2 => 'CLERKS',
 GROUP2_CPU => 25);
end;
/
-- Note: the final output is still multi-level
begin
 DBMS_RESOURCE_MANAGER.DELETE_PLAN ('JOBS_PLAN');
end;
/

/* 11g : to implement the following plan:
 CG            Level1  Level2  Level 3
 SYS_GROUP     100%
 MYGROUP1              80%
 MYGROUP2              20%
 OTHER_GROUPS                  100%
*/
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_SIMPLE_PLAN(SIMPLE_PLAN => 'SIMPLE_PLAN1',
 CONSUMER_GROUP1 => 'MYGROUP1', GROUP1_PERCENT => 80,
 CONSUMER_GROUP2 => 'MYGROUP2', GROUP2_PERCENT => 20);
END;

-- *****************
/*  Complex Plan: CG, Resource P., Res. Plan Dir. */

-- create resource plan (10g)
```

```
begin
 dbms_resource_manager.create_plan(
  PLAN    => 'NW_PLAN',
  COMMENT => 'New World Plan',
  CPU_MTH => 'EMPHASIS');  -- or RATIO
end;
/

-- create resource plan 11g
-- MGMT_MTH for specifying how much CPU each consumer group or subplan gets.
--  'EMPHASIS', the default method, is for single-level or multilevel plans
--   that use percentages. 'RATIO' is for single-level plans that use ratios
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PLAN(
 PLAN => 'NW_PLAN',
 COMMENT => 'New World Plan',
 MGMT_MTH => 'EMPHASIS',
 SUB_PLAN => FALSE -- If TRUE, the plan cannot be used as the top plan
 );
END;
/

select DISTINCT plan, num_plan_directives, cpu_method
 from dba_rsrc_plans;


-- Resource Plan Directives
begin
 dbms_resource_manager.create_plan_directive(
 PLAN => 'NW_PLAN',
 COMMENT => 'New World Directive',
 GROUP_OR_SUBPLAN => 'PROGRAMMERS',
 MAX_IDLE_TIME => '120');
end;
/

-- other parameters are
 CPU_Pn where n from 1 to 8 (10g)
 MGMT_Pn where n from 1 to 8 (11g)
 ACTIVE_SESS_POOL_P1
 QUEUEING_P1
 PARALLEL_DEGREE_LIMIT_P1
 SWITCH_GROUP  -- CG to switch to, also accept 'CANCEL_SQL' 'KILL_SESSION'
 SWITCH_TIME -- time (in CPU seconds) for a call before SWITCH_GROUP is taken
 SWITCH_ESTIMATE -- TRUE or FALSE.
 MAX_EST_EXEC_TIME
 UNDO_POOL -- in  KB
 MAX_IDLE_TIME
 MAX_IDLE_BLOCKER_TIME  -- in seconds
 SWITCH_TIME_IN_CALL -- (10g)
 SWITCH_FOR_CALL  -- (11g)
 SWITCH_IO_MEGABYTES -- (11g) max megabytes of I/O by a session
 SWITCH_IO_REQS  -- (11g) max number of I/O requests
 SWITCH_FOR_CALL -- (11g) If TRUE, a session that was automatically switched
                 --  to another consumer group (according to SWITCH_TIME,
                 --  SWITCH_IO_MEGABYTES, or SWITCH_IO_REQS) is returned to
                 --  its original consumer group when the top level call
```

```
                  --  completes.
-- examples in 11g
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (
 PLAN => 'DAYTIME',
 GROUP_OR_SUBPLAN => 'REPORTING',
 COMMENT => 'Reporting group',
 MGMT_P1 => 15,
 PARALLEL_DEGREE_LIMIT_P1 => 8,
 ACTIVE_SESS_POOL_P1 => 4);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (
 PLAN => 'DAYTIME',
 GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
 COMMENT => 'This one is required',
 MGMT_P1 => 10);
END;
/

-- Creating Multi-Level Plan Directives
-- 10g
-- 11g: replace CPU_Pn with MGMT_Pn
begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'NW_PLAN',
 COMMENT => 'SYSTEM USERS',
 GROUP_OR_SUBPLAN => 'SYS_GROUP',
 CPU_P1=> 100);
end;
/

begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'NW_PLAN',
 COMMENT => 'Clerks CPU quota',
 GROUP_OR_SUBPLAN => 'CLERKS',
 CPU_P2=> 30);
end;
/

BEGIN
dbms_resource_manager.create_plan_directive(
 PLAN => 'NW_PLAN',
 COMMENT => 'Progs CPU quota',
 GROUP_OR_SUBPLAN => 'PROGRAMMERS',
 CPU_P2=> 70);
end;
/

begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'NW_PLAN',
 COMMENT => 'OTHER_GROUPS CPU ALLOCATION',
 GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
 CPU_P3=> 100);
end;
/
```

```
-- Creating Automatic Consumer Group Switching Directives
begin
 dbms_resource_manager.create_plan_directive(
 PLAN => 'NW_PLAN',
 COMMENT => 'LIMIT CLERKS EXECUTION TIME',
 GROUP_OR_SUBPLAN => 'CLERKS',
 SWITCH_GROUP => 'CANCEL_SQL',
 SWITCH_TIME => 3600); -- one hour
end;
/

begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'NW_PLAN',
 COMMENT => 'SWITCH PROGRAMMERS TEMPORARILY',
 GROUP_OR_SUBPLAN => 'PROGRAMMERS',
 SWITCH_TIME_IN_CALL => 900,
 SWITCH_GROUP => 'LOW_GROUP',
 SWITCH_ESTIMATE => TRUE);
end;
/

-- Updating Resource Plan Directives
begin
dbms_resource_manager.update_plan_directive(
 PLAN => 'NW_PLAN',
 GROUP_OR_SUBPLAN => 'PROGRAMMERS',
 NEW_SWITCH_ESTIMATE => FALSE);
end;
/

-- Deleting Resource Plan Directives
begin
 dbms_resource_manager.delete_plan_directive(
 PLAN => 'NW_PLAN',
 GROUP_OR_SUBPLAN => 'PROGRAMMERS');
end;
/


-- validating Pending Area
exec dbms_resource_manager.validate_pending_area;

-- submitting Pending Area
exec dbms_resource_manager.submit_pending_area;



-- Enabling the Database Resource Manager
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = NW_PLAN;

-- Switching Resource Consumer Groups
-- for a session
begin
 dbms_resource_manager.switch_consumer_group_for_sess (
  SESSION_ID => '145',
```

```
  SESSION_SERIAL=> '49',
  CONSUMER_GROUP => 'SYS_GROUP');
end;
/

-- for all sessions of a specific user
begin
 dbms_resource_manager.switch_consumer_group_for_user (
 USER => 'HR',
 CONSUMER_GROUP => 'LOW_GROUP');
end;
/


-- switch C Groups by client
-- require priv
begin
 dbms_resource_manager_privs.grant_switch_consumer_group(
  GRANTEE_NAME => 'U1',  -- a role name can be given as well
  CONSUMER_GROUP => 'PROGRAMMERS',
  GRANT_OPTION => FALSE);
end;
/

-- Enabling Users or Applications to Manually Switch Consumer Groups
DECLARE
 original_group varchar2(30);
 dummy varchar2(30);
BEGIN
 DBMS_SESSION.SWITCH_CURRENT_CONSUMER_GROUP(
  'MARKETING', original_group, FALSE);
 --  execute some SQL
 null;
 DBMS_SESSION.SWITCH_CURRENT_CONSUMER_GROUP(
  original_group, dummy, FALSE);
END;
/

-- Granting and Revoking the Switch Privilege
BEGIN
-- GRANT_.. or REVOKE_..
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP(
 GRANTEE_NAME => 'SCOTT',
 CONSUMER_GROUP => 'OLTP',
 GRANT_OPTION => false);
END;
/

-- Enabling Oracle Database Resource Manager and Switching Plans
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'mydb_plan';
-- the resource plan can be changed only by setting
--  the RESOURCE_MANAGER_PLAN initialization parameter.
--  It disallows changes by Window opening.
--  This restriction can be lifted by rerunning the command without FORCE
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'FORCE:mydb_plan';
-- disable the Resource Manager
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = '';
```

```
-- ***************
/* Putting it all together Example implementation  */

OFF_HOURS_PLAN
       SYS_GROUP       10
       NIGHT_GROUP     5
       REPORTS_GROUP   2
       OTHER_GROUPS    1

DAY_PLAN
LEVEL1 SYS_GROUP       100%
LEVEL2 OLTP_PLAN       100%
       LEVEL1  OLTP_GROUP      90%     DREPORTS_GROUP  10%
       LEVEL2  OTHERS_GROUPS   100%
LEVEL3 OTHERS_GROUPS   100%


exec dbms_resource_manager.create_pending_area();

begin
 dbms_resource_manager.create_consumer_group('OLTP_GROUP','Incoming orders');
end;
/

begin
 dbms_resource_manager.create_consumer_group('DREPORTS_GROUP','DAYTIME
REPORTS');
end;
/

begin
 dbms_resource_manager.create_consumer_group('NIGHT_GROUP','BULK LOADS');
end;
/

begin
 dbms_resource_manager.create_consumer_group('REPORTS_GROUP','OFF HOURS
REPORTS');
end;
/

-- Creating the Resource Plans
begin
 dbms_resource_manager.create_plan(
 PLAN => 'DAY_PLAN',
 COMMENT => 'NORMAL WORKING HOURS ');
end;
/

begin
 dbms_resource_manager.create_plan
 (PLAN => 'OLTP_PLAN', COMMENT => 'ORDER ENTRY SUB-PLAN');
end;
/
```

```
begin
dbms_resource_manager.create_plan(
 PLAN => 'OFF_HOURS_PLAN',
 COMMENT => 'GOVERNS NON-WORKING HOURS',
 CPU_MTH => 'RATIO');
end;
/

-- PLAN DIRECTIVE
begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'OFF_HOURS_PLAN',
 GROUP_OR_SUBPLAN => 'SYS_GROUP',
 COMMENT => 'CPU ALLOCATION FOR SYS_GROUP',
 CPU_P1 => 10);
end;
/

begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'OFF_HOURS_PLAN',
 GROUP_OR_SUBPLAN => 'NIGHT_GROUP',
 COMMENT => 'CPU ALLOCATION FOR NIGHTLY JOBS',
 CPU_P1 => 5);
end;
/


begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'OFF_HOURS_PLAN',
 GROUP_OR_SUBPLAN => 'REPORTS_GROUP',
 COMMENT => 'CPU ALLOCATION FOR NIGHTLY REPORTS',
 CPU_P1 => 2);
end;
/

begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'OFF_HOURS_PLAN',
 GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
 COMMENT => 'CPU ALLOCATION FOR OTHER_GROUPS',
 CPU_P1 => 1);
end;
/


-- Creating the OLTP_PLAN Plan Directives
begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'OLTP_PLAN',
 GROUP_OR_SUBPLAN => 'OLTP_GROUP',
 COMMENT => 'CPU ALLOCATION FOR OLTP USERS',
 CPU_P1 => 90);
end;
/
```

```
begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'OLTP_PLAN',
 GROUP_OR_SUBPLAN => 'DREPORTS_GROUP',
 COMMENT => 'CPU ALLOCATION FOR DAYTIME REPORTING',
 CPU_P1 => 10);
end;
/

begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'OLTP_PLAN',
 GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
 COMMENT => 'CPU ALLOCATION FOR OTHER_GROUPS',
 CPU_P2 => 100);
end;
/


-- DAY_PLAN Plan Directives
begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'DAY_PLAN',
 GROUP_OR_SUBPLAN => 'SYS_GROUP',
 COMMENT => 'CPU ALLOCATION FOR SYS_GROUP',
 CPU_P1 => 100);
end;
/


begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'DAY_PLAN',
 GROUP_OR_SUBPLAN => 'OLTP_PLAN',
 COMMENT => 'CPU ALLOCATION FOR OLTP_PLAN SUB-PLAN',
 CPU_P2 => 100);
end;
/

begin
dbms_resource_manager.create_plan_directive(
 PLAN => 'DAY_PLAN',
 GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
 COMMENT => 'CPU ALLOCATION FOR OTHER_GROUPS',
 CPU_P3 => 100);
end;
/

exec dbms_resource_manager.validate_pending_area;

exec dbms_resource_manager.submit_pending_area;


-- Enabling the Resource Plans
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'DAY_PLAN';
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'OFF_HOURS_PLAN';
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = '';
```

```
-- testing

exec dbms_resource_manager.create_pending_area();
begin
 dbms_resource_manager.set_consumer_group_mapping(
       ATTRIBUTE => dbms_resource_manager.oracle_user,
       VALUE => 'U1',
       CONSUMER_GROUP => 'NIGHT_GROUP');
end;
/

begin
 dbms_resource_manager_privs.grant_switch_consumer_group(
  GRANTEE_NAME => 'U1',  -- a role name can be given as well
  CONSUMER_GROUP => 'NIGHT_GROUP',
  GRANT_OPTION => FALSE);
end;
/

begin
 dbms_resource_manager_privs.grant_switch_consumer_group(
  GRANTEE_NAME => 'U2',
  CONSUMER_GROUP => 'REPORTS_GROUP',
  GRANT_OPTION => FALSE);
end;
/

begin
 dbms_resource_manager.set_consumer_group_mapping(
       ATTRIBUTE => dbms_resource_manager.oracle_user, -- it is constant (no
quots)
       VALUE => 'U2',
       CONSUMER_GROUP => 'REPORTS_GROUP');
end;
/

begin
 dbms_resource_manager_privs.grant_switch_consumer_group(
  GRANTEE_NAME => 'U3',
  CONSUMER_GROUP => 'OLTP_GROUP',
  GRANT_OPTION => FALSE);
end;
/

begin
 dbms_resource_manager.set_consumer_group_mapping(
       ATTRIBUTE => dbms_resource_manager.oracle_user,
       VALUE => 'U3',
       CONSUMER_GROUP => 'OLTP_GROUP');
end;
/

exec dbms_resource_manager.submit_pending_area;

SET SERVEROUTPUT ON
```

```
DECLARE
 D DATE := SYSDATE;
 DUMMY NUMBER;
BEGIN
 DBMS_OUTPUT.PUT_LINE('Start Time:'|| to_char(D,'hh24:mi:ss'));
 FOR I IN 1..3000000 LOOP
  DUMMY := SQRT(DBMS_RANDOM.VALUE(1,8100));
 END LOOP;
 DBMS_OUTPUT.PUT_LINE('End Time:'|| to_char(sysdate,'hh24:mi:ss'));
 DBMS_OUTPUT.PUT_LINE('Elapsed :'|| round((sysdate-d)*24*60*60,2) || ' s' );
END;
/
```

Obtaining Information on Database Resource Manager

```
-- resource consumer groups that exist in the database
select
CONSUMER_GROUP_ID,CONSUMER_GROUP,CPU_METHOD,MGMT_METHOD,INTERNAL_USE,COMMENTS
,CATEGORY,STATUS,MANDATORY
from DBA_RSRC_CONSUMER_GROUPS;

-- resource directives
select
PLAN,GROUP_OR_SUBPLAN,TYPE,CPU_P1,CPU_P2,CPU_P3,CPU_P4,CPU_P5,CPU_P6,CPU_P7,C
PU_P8,MGMT_P1,MGMT_P2,MGMT_P3,MGMT_P4,MGMT_P5,MGMT_P6,MGMT_P7,MGMT_P8,ACTIVE_
SESS_POOL_P1,QUEUEING_P1,PARALLEL_DEGREE_LIMIT_P1,SWITCH_GROUP,SWITCH_FOR_CAL
L,SWITCH_TIME,SWITCH_IO_MEGABYTES,SWITCH_IO_REQS,SWITCH_ESTIMATE,MAX_EST_EXEC
_TIME,UNDO_POOL,MAX_IDLE_TIME,MAX_IDLE_BLOCKER_TIME,SWITCH_TIME_IN_CALL,COMME
NTS,STATUS,MANDATORY
from DBA_RSRC_PLAN_DIRECTIVES;

-- the groups to which a user or role is allowed to belong or be switched
select GRANTEE,GRANTED_GROUP,GRANT_OPTION,INITIAL_GROUP
from DBA_RSRC_CONSUMER_GROUP_PRIVS

-- Viewing Plan Information
select
PLAN_ID,PLAN,NUM_PLAN_DIRECTIVES,CPU_METHOD,MGMT_METHOD,ACTIVE_SESS_POOL_MTH,
PARALLEL_DEGREE_LIMIT_MTH,QUEUEING_MTH,SUB_PLAN,COMMENTS,STATUS,MANDATORY
from DBA_RSRC_PLANS

-- Viewing Current Consumer Groups for Sessions
SELECT SID,SERIAL#,USERNAME,RESOURCE_CONSUMER_GROUP FROM V$SESSION;

-- display the current top level plan and all of its descendent subplans
SELECT NAME, IS_TOP_PLAN FROM V$RSRC_PLAN;

-- mapping pairs for all of the session attributes
select ATTRIBUTE,VALUE,CONSUMER_GROUP,STATUS
from DBA_RSRC_GROUP_MAPPINGS;

-- current mapping priority of each attribute
select ATTRIBUTE,PRIORITY,STATUS
from DBA_RSRC_MAPPING_PRIORITY;

-- historical information on resource plan activation: snapshots of
V$RSRC_PLAN_HISTORY
```

```
select
SNAP_ID,DBID,INSTANCE_NUMBER,SEQUENCE#,START_TIME,END_TIME,PLAN_ID,PLAN_NAME,
CPU_MANAGED
from DBA_HIST_RSRC_PLAN;

-- historical statistical information on consumer groups. snapshots of:
V$RSRC_CONS_GROUP_HISTORY
select
SNAP_ID,DBID,INSTANCE_NUMBER,SEQUENCE#,CONSUMER_GROUP_ID,CONSUMER_GROUP_NAME,
REQUESTS,CPU_WAIT_TIME,CPU_WAITS,CONSUMED_CPU_TIME,YIELDS,ACTIVE_SESS_LIMIT_H
IT,UNDO_LIMIT_HIT,SWITCHES_IN_CPU_TIME,SWITCHES_OUT_CPU_TIME,SWITCHES_IN_IO_M
EGABYTES,SWITCHES_OUT_IO_MEGABYTES,SWITCHES_IN_IO_REQUESTS,SWITCHES_OUT_IO_RE
QUESTS,SQL_CANCELED,ACTIVE_SESS_KILLED,IDLE_SESS_KILLED,IDLE_BLKR_SESS_KILLED
,QUEUED_TIME,QUEUE_TIME_OUTS,IO_SERVICE_TIME,IO_SERVICE_WAITS,SMALL_READ_MEGA
BYTES,SMALL_WRITE_MEGABYTES,LARGE_READ_MEGABYTES,LARGE_WRITE_MEGABYTES,SMALL_
READ_REQUESTS,SMALL_WRITE_REQUESTS,LARGE_READ_REQUESTS,LARGE_WRITE_REQUESTS
from DBA_HIST_RSRC_CONSUMER_GROUP ;

-- for each consumer group in the plan showing the cumulative statistics for
the consumer group.
-- sequence# as as in V$RSRC_PLAN_HISTORY
select
SEQUENCE#,ID,NAME,REQUESTS,CPU_WAIT_TIME,CPU_WAITS,CONSUMED_CPU_TIME,YIELDS,A
CTIVE_SESS_LIMIT_HIT,UNDO_LIMIT_HIT,SWITCHES_IN_CPU_TIME,SWITCHES_OUT_CPU_TIM
E,SWITCHES_IN_IO_MEGABYTES,SWITCHES_OUT_IO_MEGABYTES,SWITCHES_IN_IO_REQUESTS,
SWITCHES_OUT_IO_REQUESTS,SQL_CANCELED,ACTIVE_SESS_KILLED,IDLE_SESS_KILLED,IDL
E_BLKR_SESS_KILLED,QUEUED_TIME,QUEUE_TIME_OUTS,IO_SERVICE_TIME,IO_SERVICE_WAI
TS,SMALL_READ_MEGABYTES,SMALL_WRITE_MEGABYTES,LARGE_READ_MEGABYTES,LARGE_WRIT
E_MEGABYTES,SMALL_READ_REQUESTS,SMALL_WRITE_REQUESTS,LARGE_READ_REQUESTS,LARG
E_WRITE_REQUESTS
from V$RSRC_CONS_GROUP_HISTORY;

select
SEQUENCE#,ID,NAME,REQUESTS,CPU_WAIT_TIME,CPU_WAITS,CONSUMED_CPU_TIME,YIELDS,A
CTIVE_SESS_LIMIT_HIT,UNDO_LIMIT_HIT,SWITCHES_IN_CPU_TIME,SWITCHES_OUT_CPU_TIM
E,SWITCHES_IN_IO_MEGABYTES,SWITCHES_OUT_IO_MEGABYTES,SWITCHES_IN_IO_REQUESTS,
SWITCHES_OUT_IO_REQUESTS,SQL_CANCELED,ACTIVE_SESS_KILLED,IDLE_SESS_KILLED,IDL
E_BLKR_SESS_KILLED,QUEUED_TIME,QUEUE_TIME_OUTS,IO_SERVICE_TIME,IO_SERVICE_WAI
TS,SMALL_READ_MEGABYTES,SMALL_WRITE_MEGABYTES,LARGE_READ_MEGABYTES,LARGE_WRIT
E_MEGABYTES,SMALL_READ_REQUESTS,SMALL_WRITE_REQUESTS,LARGE_READ_REQUESTS,LARG
E_WRITE_REQUESTS
from V$RSRC_CONS_GROUP_HISTORY;

-- resource usage and stats data on currently active resource consumer groups
select
ID,NAME,ACTIVE_SESSIONS,EXECUTION_WAITERS,REQUESTS,CPU_WAIT_TIME,CPU_WAITS,CO
NSUMED_CPU_TIME,YIELDS,QUEUE_LENGTH,CURRENT_UNDO_CONSUMPTION,ACTIVE_SESSION_L
IMIT_HIT,UNDO_LIMIT_HIT,SWITCHES_IN_CPU_TIME,SWITCHES_OUT_CPU_TIME,SWITCHES_I
N_IO_MEGABYTES,SWITCHES_OUT_IO_MEGABYTES,SWITCHES_IN_IO_REQUESTS,SWITCHES_OUT
_IO_REQUESTS,SQL_CANCELED,ACTIVE_SESSIONS_KILLED,IDLE_SESSIONS_KILLED,IDLE_BL
KR_SESSIONS_KILLED,QUEUED_TIME,QUEUE_TIME_OUTS,IO_SERVICE_TIME,IO_SERVICE_WAI
TS,SMALL_READ_MEGABYTES,SMALL_WRITE_MEGABYTES,LARGE_READ_MEGABYTES,LARGE_WRIT
E_MEGABYTES,SMALL_READ_REQUESTS,SMALL_WRITE_REQUESTS,LARGE_READ_REQUESTS,LARG
E_WRITE_REQUESTS,V$RSRC_CONSUMER_GROUP
FROM v$rsrc_consumer_group;

-- a history of resources consumed and cumulative CPU wait time (due to
resource management) per consumer group for the past minute
select
BEGIN_TIME,END_TIME,INTSIZE_CSEC,SEQUENCE#,CONSUMER_GROUP_ID,CONSUMER_GROUP_N
AME,CPU_CONSUMED_TIME,CPU_WAIT_TIME,IO_REQUESTS,IO_MEGABYTES
```

```
from V$RSRCMGRMETRIC;

-- when resource plans were enabled or disabled on the instance
SELECT sequence# seq, name plan_name,
 to_char(start_time, 'DD-MON-YY HH24:MM') start_time,
 to_char(end_time, 'DD-MON-YY HH24:MM') end_time, window_name
FROM v$rsrc_plan_history;

-- how the session has been affected by the Resource Manager
select
SID,CURRENT_CONSUMER_GROUP_ID,ORIG_CONSUMER_GROUP_ID,MAPPING_ATTRIBUTE,MAPPED
_CONSUMER_GROUP,STATE,ACTIVE,CURRENT_IDLE_TIME,CURRENT_CPU_WAIT_TIME,CPU_WAIT
_TIME,CURRENT_CPU_WAITS,CPU_WAITS,CURRENT_CONSUMED_CPU_TIME,CONSUMED_CPU_TIME
,CURRENT_ACTIVE_TIME,ACTIVE_TIME,CURRENT_QUEUED_TIME,QUEUED_TIME,CURRENT_YIEL
DS,YIELDS,CURRENT_UNDO_CONSUMPTION,MAX_UNDO_CONSUMPTION,SQL_CANCELED,QUEUE_TI
ME_OUTS,ESTIMATED_EXECUTION_LIMIT_HIT,CURRENT_IO_SERVICE_TIME,IO_SERVICE_TIME
,CURRENT_IO_SERVICE_WAITS,IO_SERVICE_WAITS,CURRENT_SMALL_READ_MEGABYTES,SMALL
_READ_MEGABYTES,CURRENT_LARGE_READ_MEGABYTES,LARGE_READ_MEGABYTES,CURRENT_SMA
LL_WRITE_MEGABYTES,SMALL_WRITE_MEGABYTES,CURRENT_LARGE_WRITE_MEGABYTES,LARGE_
WRITE_MEGABYTES,CURRENT_SMALL_READ_REQUESTS,SMALL_READ_REQUESTS,CURRENT_SMALL
_WRITE_REQUESTS,SMALL_WRITE_REQUESTS,CURRENT_LARGE_READ_REQUESTS,LARGE_READ_R
EQUESTS,CURRENT_LARGE_WRITE_REQUESTS,LARGE_WRITE_REQUESTS
from v$rsrc_session_info;
```

Monitoring Oracle Database Resource Manager

```
-- display the currently active resource plan and its subplans.
SELECT name, is_top_plan FROM v$rsrc_plan;

-- resource usage and stats data on currently active resource consumer groups
select
ID,NAME,ACTIVE_SESSIONS,EXECUTION_WAITERS,REQUESTS,CPU_WAIT_TIME,CPU_WAITS,CO
NSUMED_CPU_TIME,YIELDS,QUEUE_LENGTH,CURRENT_UNDO_CONSUMPTION,ACTIVE_SESSION_L
IMIT_HIT,UNDO_LIMIT_HIT,SWITCHES_IN_CPU_TIME,SWITCHES_OUT_CPU_TIME,SWITCHES_I
N_IO_MEGABYTES,SWITCHES_OUT_IO_MEGABYTES,SWITCHES_IN_IO_REQUESTS,SWITCHES_OUT
_IO_REQUESTS,SQL_CANCELED,ACTIVE_SESSIONS_KILLED,IDLE_SESSIONS_KILLED,IDLE_BL
KR_SESSIONS_KILLED,QUEUED_TIME,QUEUE_TIME_OUTS,IO_SERVICE_TIME,IO_SERVICE_WAI
TS,SMALL_READ_MEGABYTES,SMALL_WRITE_MEGABYTES,LARGE_READ_MEGABYTES,LARGE_WRIT
E_MEGABYTES,SMALL_READ_REQUESTS,SMALL_WRITE_REQUESTS,LARGE_READ_REQUESTS,LARG
E_WRITE_REQUESTS,V$RSRC_CONSUMER_GROUP
FROM v$rsrc_consumer_group;

-- how the session has been affected by the Resource Manager
select
SID,CURRENT_CONSUMER_GROUP_ID,ORIG_CONSUMER_GROUP_ID,MAPPING_ATTRIBUTE,MAPPED
_CONSUMER_GROUP,STATE,ACTIVE,CURRENT_IDLE_TIME,CURRENT_CPU_WAIT_TIME,CPU_WAIT
_TIME,CURRENT_CPU_WAITS,CPU_WAITS,CURRENT_CONSUMED_CPU_TIME,CONSUMED_CPU_TIME
,CURRENT_ACTIVE_TIME,ACTIVE_TIME,CURRENT_QUEUED_TIME,QUEUED_TIME,CURRENT_YIEL
DS,YIELDS,CURRENT_UNDO_CONSUMPTION,MAX_UNDO_CONSUMPTION,SQL_CANCELED,QUEUE_TI
ME_OUTS,ESTIMATED_EXECUTION_LIMIT_HIT,CURRENT_IO_SERVICE_TIME,IO_SERVICE_TIME
,CURRENT_IO_SERVICE_WAITS,IO_SERVICE_WAITS,CURRENT_SMALL_READ_MEGABYTES,SMALL
_READ_MEGABYTES,CURRENT_LARGE_READ_MEGABYTES,LARGE_READ_MEGABYTES,CURRENT_SMA
LL_WRITE_MEGABYTES,SMALL_WRITE_MEGABYTES,CURRENT_LARGE_WRITE_MEGABYTES,LARGE_
WRITE_MEGABYTES,CURRENT_SMALL_READ_REQUESTS,SMALL_READ_REQUESTS,CURRENT_SMALL
_WRITE_REQUESTS,SMALL_WRITE_REQUESTS,CURRENT_LARGE_READ_REQUESTS,LARGE_READ_R
EQUESTS,CURRENT_LARGE_WRITE_REQUESTS,LARGE_WRITE_REQUESTS
from v$rsrc_session_info;

SELECT se.sid sess_id, co.name consumer_group,
```

```
se.state, se.consumed_cpu_time cpu_time, se.cpu_wait_time, se.queued_time
FROM v$rsrc_session_info se, v$rsrc_consumer_group co
WHERE se.current_consumer_group_id = co.id;

-- when resource plans were enabled or disabled on the instance
SELECT sequence# seq, name plan_name,
to_char(start_time, 'DD-MON-YY HH24:MM') start_time,
to_char(end_time, 'DD-MON-YY HH24:MM') end_time, window_name
FROM v$rsrc_plan_history;

-- how resources were shared among the consumer groups over time
select sequence# seq, name, cpu_wait_time, cpu_waits,
consumed_cpu_time from V$RSRC_CONS_GROUP_HISTORY;

-- sequence# as as in V$RSRC_PLAN_HISTORY
select
SEQUENCE#,ID,NAME,REQUESTS,CPU_WAIT_TIME,CPU_WAITS,CONSUMED_CPU_TIME,YIELDS,A
CTIVE_SESS_LIMIT_HIT,UNDO_LIMIT_HIT,SWITCHES_IN_CPU_TIME,SWITCHES_OUT_CPU_TIM
E,SWITCHES_IN_IO_MEGABYTES,SWITCHES_OUT_IO_MEGABYTES,SWITCHES_IN_IO_REQUESTS,
SWITCHES_OUT_IO_REQUESTS,SQL_CANCELED,ACTIVE_SESS_KILLED,IDLE_SESS_KILLED,IDL
E_BLKR_SESS_KILLED,QUEUED_TIME,QUEUE_TIME_OUTS,IO_SERVICE_TIME,IO_SERVICE_WAI
TS,SMALL_READ_MEGABYTES,SMALL_WRITE_MEGABYTES,LARGE_READ_MEGABYTES,LARGE_WRIT
E_MEGABYTES,SMALL_READ_REQUESTS,SMALL_WRITE_REQUESTS,LARGE_READ_REQUESTS,LARG
E_WRITE_REQUESTS
from V$RSRC_CONS_GROUP_HISTORY;
```

Using Oracle Scheduler

Using Jobs

- For Batch Files in Windows:
  - The OracleJobscheduler service must exist in a running state
  - The user who runs this service should have the Logon as batch job privilege
  - .bat file cannot be run directly, but should be called as an argument of cmd.exe

```
-- creating a job
begin
 dbms_scheduler.create_job (
 job_name => 'Calc_Job',
 job_type => 'stored_procedure',  -- stored_procedure, plsql_block,
executable, chain
 job_action => 'POPULATE_DEPT_SALS',
 start_date => SYSDATE, -- or to_date('01-08-2006 17:46','dd-mm-yyyy
hh24:mi'),
 repeat_interval => 'FREQ=MINUTELY; INTERVAL=2',
 comments => 'Annual Department Salaries');
end;
/

BEGIN
DBMS_SCHEDULER.CREATE_JOB (
 job_name => 'update_sales',
 job_type => 'STORED_PROCEDURE',
 job_action => 'OPS.SALES_PKG.UPDATE_SALES_SUMMARY',
 start_date => '28-APR-03 07.00.00 PM Australia/Sydney',
 repeat_interval => 'FREQ=DAILY;INTERVAL=2', /* every other day */
 end_date => '20-NOV-04 07.00.00 PM Australia/Sydney',
 job_class => 'batch_update_jobs',
 comments => 'My new job');
END;
/

-- for SYSTEM=TRUE: automatically created by Oracle jobs
select
OWNER,JOB_NAME,JOB_SUBNAME,JOB_STYLE,JOB_CREATOR,CLIENT_ID,GLOBAL_UID,PROGRAM
_OWNER,PROGRAM_NAME,JOB_TYPE,JOB_ACTION,NUMBER_OF_ARGUMENTS,SCHEDULE_OWNER,SC
HEDULE_NAME,SCHEDULE_TYPE,START_DATE,REPEAT_INTERVAL,EVENT_QUEUE_OWNER,EVENT_
QUEUE_NAME,EVENT_QUEUE_AGENT,EVENT_CONDITION,EVENT_RULE,END_DATE,JOB_CLASS,EN
ABLED,AUTO_DROP,RESTARTABLE,STATE,JOB_PRIORITY,RUN_COUNT,MAX_RUNS,FAILURE_COU
NT,MAX_FAILURES,RETRY_COUNT,LAST_START_DATE,LAST_RUN_DURATION,NEXT_RUN_DATE,S
CHEDULE_LIMIT,MAX_RUN_DURATION,LOGGING_LEVEL,STOP_ON_WINDOW_CLOSE,INSTANCE_ST
ICKINESS,RAISE_EVENTS,SYSTEM,JOB_WEIGHT,NLS_ENV,SOURCE,DESTINATION,CREDENTIAL
_OWNER,CREDENTIAL_NAME,INSTANCE_ID,DEFERRED_DROP,COMMENTS,FLAGS
from dba_SCHEDULER_JOBS;


-- Setting Repeat Intervals
FREQ YEARLY, MONTHLY, WEEKLY, DAILY, HOURLY, MINUTELY, and SECONDLY.
INTERVAL a number of maximum 999
BYMONTH 1-12   1,4,6
BYYEARDAY any positive or negative number
BYMONTHDAY any positive or negative number (eg -1 last day of the month)
```

```
BYDAY (MON, TUE, and so on) can be prefixed with a number -1FRI
BYHOUR 0-23
BYMINUTE 0-59
BYSECOND 0-59

Every Monday           FREQ=WEEKLY; BYDAY=MON;
Every other Monday     FREQ=WEEKLY; BYDAY=MON; INTERVAL=2;
Last day of each month         FREQ=MONTHLY; BYMONTHDAY=-1;
Every January 7        FREQ=YEARLY; BYMONTH=JAN; BYMONTHDAY=7;
Second Wednesday of    FREQ=MONTHLY; BYDAY=2WED;
each month
Every hour             FREQ=HOURLY;
Every 4 hours          FREQ=HOURLY; INTERVAL=4;
Hourly on the first    FREQ=HOURLY; BYMONTHDAY=1;
day of each month
15th day of every other FREQ=MONTHLY; BYMONTHDAY=15; INTERVAL=2
month
-- MORE EXAMPLES
Daily at 4, 5, and 6PM: FREQ=DAILY; BYHOUR=16,17,18;
15th day of every other month: FREQ=MONTHLY; INTERVAL=2; BYMONTHDAY=15;
29th day of every month: FREQ=MONTHLY; BYMONTHDAY=29;
Second Wednesday of each month: FREQ=MONTHLY; BYDAY=2WED;
Last Friday of the year: FREQ=YEARLY; BYDAY=-1FRI;
Every 50 hours: FREQ=HOURLY; INTERVAL=50;
Last day of every other month: FREQ=MONTHLY; INTERVAL=2; BYMONTHDAY=-1;
Hourly for the first three days of every month: FREQ=HOURLY;
BYMONTHDAY=1,2,3;
Last workday of every month (assuming that workdays are Monday through
Friday):
 FREQ=MONTHLY; BYDAY=MON,TUE,WED,THU,FRI; BYSETPOS=-1
Last workday of every month, excluding company holidays. (This example
references an existing named schedule called Company_Holidays):
 FREQ=MONTHLY; BYDAY=MON,TUE,WED,THU,FRI; EXCLUDE=Company_Holidays;
BYSETPOS=-1
Noon every Friday and on company holidays:
 FREQ=YEARLY;BYDAY=FRI;BYHOUR=12;INCLUDE=Company_Holidays

-- Testing Repeat Intervals
set serveroutput on
DECLARE
  start_date TIMESTAMP;
  return_date_after TIMESTAMP;
  next_run_date TIMESTAMP;
BEGIN
  start_date := to_timestamp('01-01-2006 00:00:00','DD-MM-YYYY HH24:MI:SS');
  DBMS_SCHEDULER.EVALUATE_CALENDAR_STRING(
   CALENDAR_STRING   => 'FREQ=MONTHLY;INTERVAL=1',
   START_DATE       => start_date,
   RETURN_DATE_AFTER => null,
   NEXT_RUN_DATE     => next_run_date);
 DBMS_OUTPUT.PUT_LINE('next_run_date: ' || next_run_date);
END;
/

alter session set nls_date_language=English
/

DECLARE
```

```
 start_date       TIMESTAMP;
 return_date_after TIMESTAMP;
 next_run_date            TIMESTAMP;
BEGIN
 start_date := to_timestamp('10-10-2004 10:00:00','DD-MM-YYYY HH24:MI:SS');
 return_date_after := start_date;
 FOR i IN 1..10 LOOP
  DBMS_SCHEDULER.EVALUATE_CALENDAR_STRING(
   'FREQ=MONTHLY; INTERVAL=2; BYMONTHDAY=15',
   start_date, return_date_after, next_run_date);
  DBMS_OUTPUT.PUT_LINE('Next Run Date: ' || to_char(next_run_date,'Dy dd-MM-
RR HH24:MI'));
  return_date_after := next_run_date;
 END LOOP;
END;
/


-- Creating a Set of Regular Jobs
DECLARE
newjob sys.job;
newjobarr sys.job_array;
BEGIN
-- Create an array of JOB object types
newjobarr := sys.job_array();
-- Allocate sufficient space in the array
newjobarr.extend(5);
-- Add definitions for 5 jobs
FOR i IN 1..5 LOOP
 -- Create a JOB object type
 newjob := sys.job(job_name => 'TESTJOB' || to_char(i),
  job_style => 'REGULAR',
  job_template => 'PROG1',
  repeat_interval => 'FREQ=MINUTELY;INTERVAL_15',
  start_date => systimestamp + interval '600' second,
  max_runs => 2,
  auto_drop => FALSE,
  enabled _> TRUE );
 -- Add it to the array
 newjobarr(i) := newjob;
END LOOP;
-- Call CREATE_JOBS to create jobs in one transaction
DBMS_SCHEDULER.CREATE_JOBS(newjobarr, 'TRANSACTIONAL');
END;
/


-- Creating a Set of Lightweight Jobs
DECLARE
 newjob sys.job;
 newjobarr sys.job_array;
BEGIN
 newjobarr := sys.job_array();
 newjobarr.extend(5);
FOR i IN 1..5 LOOP
 newjob := sys.job(job_name => 'lwjob_' || to_char(i),
  job_style => 'LIGHTWEIGHT',
```

```
   job_template => 'PROG1',
   repeat_interval => 'FREQ=MINUTELY;INTERVAL=15',
   start_date => systimestamp + interval '10' second,
   enabled => TRUE);
 newjobarr(i) := newjob;
end loop;
DBMS_SCHEDULER.CREATE_JOBS(newjobarr, 'TRANSACTIONAL');
END;
/


-- Creating Jobs Using a Named Program
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
 job_name => 'my_new_job1',
 program_name => 'my_saved_program',
 repeat_interval => 'FREQ=DAILY;BYHOUR=12',
comments => 'Daily at noon');
END;
/


-- Creating Jobs Using a Named Schedule
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
 job_name => 'my_new_job2',
 job_type => 'PLSQL_BLOCK',
 job_action => 'BEGIN SALES_PKG.UPDATE_SALES_SUMMARY; END;',
 schedule_name => 'my_saved_schedule');
END;
/

-- Creating Jobs Using a Named Program and Schedule
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
 job_name => 'my_new_job3',
 program_name => 'my_saved_program1',
 schedule_name => 'my_saved_schedule1');
END;
/

-- Creating Remote External Jobs
-- Note: check section "Enabling and Disabling Remote External Jobs"
BEGIN
DBMS_SCHEDULER.CREATE_JOB(
 job_name => 'CLEANLOGS',
 job_type => 'EXECUTABLE',
 job_action => '/home/logowner/cleanlogs',
 repeat_interval => 'FREQ=DAILY; BYHOUR=23',
 enabled => FALSE);
DBMS_SCHEDULER.SET_ATTRIBUTE('CLEANLOGS', 'credential_name', 'LOGOWNER');
DBMS_SCHEDULER.SET_ATTRIBUTE('CLEANLOGS', 'destination', 'app455:12345');
DBMS_SCHEDULER.ENABLE('CLEANLOGS');
END;
/
-- in multiple hosts
declare
```

```
 job_prefix varchar2(30) := 'remote_';
 job_name varchar2(30);
 destinations dbms_utility.lname_array;
begin
destinations(1) := 'host1:1234';
destinations(2) := 'host2:1234';
destinations(3) := 'host3:1234';
destinations(4) := 'host4:1234';
for i in 1..destinations.LAST loop
 job_name := dbms_scheduler.generate_job_name(job_prefix);
  dbms_scheduler.create_job(job_name,
   job_type=>'executable',
   job_action=>'/u01/app/ext_backup',
   number_of_arguments=>0,
   enabled=>false);
 dbms_scheduler.set_attribute(job_name,'destination',destinations(i));
 dbms_scheduler.set_attribute(job_name,'credential_name','remote_cred');
 dbms_scheduler.enable(job_name);
end loop;
end;
/

-- run a shell script that uses SQL*Plus to submit the statements
--  external authentication is assumed
#!/bin/sh
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=orcl
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
# The following command assumes external authentication
$ORACLE_HOME/bin/sqlplus / << EOF
set serveroutput on;
select * from dual;
EXIT;
EOF

-- Copying Jobs
-- the new job is created disabled
begin
DBMS_SCHEDULER.COPY_JOB (
 old_job =>'MY_OLD_JOB',
 new_job =>'MY_NEW_JOB');
end;


-- altering jobs
-- value paramter may be  {VARCHAR2|TIMESTAMP WITH TIMEZONE|
--  PLS_INTEGER|BOOLEAN|INTERVAL DAY TO SECOND}
begin
DBMS_SCHEDULER.SET_ATTRIBUTE (
 name =>'FULL_BAKCUP',
 attribute =>'auto_drop', -- see possible values below
 value =>TRUE);
end;
/

-- attribute possible values for JOBS
auto_drop
```

```
comments
credential_name       credential to use when running an external job
database_role  Oracle Data Guard 'PRIMARY' or 'LOGICAL STANDBY'
destination           host and port on which to run a remote external job
end_date              if no value, the job repeats forever
event_spec
follow_default_timezone
instance_id
instance_stickiness   TRUE or FALSE
job_action
job_class
job_priority          the order in which jobs from that class are picked up
by
                       the job coordinator. 1(first)-5, 3 default
job_type              'PLSQL_BLOCK', 'STORED_PROCEDURE','EXECUTABLE', 'CHAIN'
job_weight            in parallel technology, 1-100
logging_level  DBMS_SCHEDULER.LOGGING_OFF, LOGGING_FAILED_RUNS,
                       LOGGING_RUNS, LOGGING_FULL
max_failures          number of times a job can fail on consecutive
max_run_duration      (INTERVAL DAY TO SECOND) if job duration exceeds this
                       value, the Scheduler raises an event of type
                       JOB_OVER_MAX_DUR. It is then up to your event handler
to
                       decide whether or not to allow the job to continue
                       max_runs the maximum number of consecutive scheduled
                       runs. disabled then.
number_of_arguments   The number of arguments if the program is inlined
                              parallel_instances     TRUE or FALSE
program_name
raise_events           job_started CONSTANT PLS_INTEGER := 1
                       job_succeeded CONSTANT PLS_INTEGER := 2
                       job_failed CONSTANT PLS_INTEGER :=4
                       job_broken CONSTANT PLS_INTEGER :=8
                       job_completed CONSTANT PLS_INTEGER :=16
                       job_stopped CONSTANT PLS_INTEGER :=32
                       job_sch_lim_reached CONSTANT PLS_INTEGER :=64
                       job_disabled CONSTANT PLS_INTEGER :=128
                       job_chain_stalled CONSTANT PLS_INTEGER :=256
                       job_all_events CONSTANT PLS_INTEGER := 511
                       job_run_completed CONSTANT PLS_INTEGER :=
job_succeeded + job_failed + job_stopped
repeat_interval       Either a PL/SQL function returning the next date
                       and time on which to run, or calendaring syntax
                       expression
restartable
schedule_limit time limit after which the job will not start
                      ( from 1 minute to 99 days )
schedule_name
start_date
stop_on_window_close


-- Setting Multiple Job Attributes for a Set of Regular Jobs
DECLARE
 newattr sys.jobattr;
 newattrarr sys.jobattr_array;
 j number;
BEGIN
```

```
-- Create new JOBATTR array
newattrarr := sys.jobattr_array();
-- Allocate enough space in the array
newattrarr.extend(20);
j := 1;
FOR i IN 1..5 LOOP
 -- Create and initialize a JOBATTR object type
 newattr := sys.jobattr(job_name => 'TESTJOB' || to_char(i),
  attr_name => 'MAX_FAILURES',
  attr_value => 5);
 -- Add it to the array.
 newattrarr(j) := newattr;
 j := j + 1;
 newattr := sys.jobattr(job_name => 'TESTJOB' || to_char(i),
  attr_name => 'COMMENTS',
  attr_value => 'Bogus comment');
 newattrarr(j) := newattr;
 j := j + 1;
 newattr := sys.jobattr(job_name => 'TESTJOB' || to_char(i),
  attr_name => 'END_DATE',
  attr_value => systimestamp + interval '24' hour);
 newattrarr(j) := newattr;
 j := j + 1;
 newattr := sys.jobattr(job_name => 'TESTJOB' || to_char(i),
  attr_name => 'SCHEDULE_LIMIT',
  attr_value => interval '1' hour);
 newattrarr(j) := newattr;
 j := j + 1;
END LOOP;
-- Call SET_JOB_ATTRIBUTES to set all 20 set attributes in one transaction
DBMS_SCHEDULER.SET_JOB_ATTRIBUTES(newattrarr, 'TRANSACTIONAL');
END;
/

-- stopping a job
-- stop job job1 and all jobs in the job class dw_jobs.
BEGIN
 DBMS_SCHEDULER.STOP_JOB('job1, sys.dw_jobs');
END;
/


-- Dropping Jobs
BEGIN
 DBMS_SCHEDULER.DROP_JOB ('job1, job3, sys.jobclass1, sys.jobclass2');
END;
/

-- Disabling Jobs
BEGIN
 DBMS_SCHEDULER.DISABLE('job1, job2, job3, sys.jobclass1');
END;
/

-- Enabling Jobs
BEGIN
 DBMS_SCHEDULER.ENABLE ('job1, job2, job3,sys.jobclass1');
```

```
END;
/
```

Using Programs

```
-- Creating Programs
BEGIN
DBMS_SCHEDULER.CREATE_PROGRAM (
 program_name => 'my_program1',
 program_action => '/usr/local/bin/date',
 program_type => 'EXECUTABLE',
comments => 'My comments here');
END;
/

BEGIN
DBMS_SCHEDULER.CREATE_PROGRAM (
 program_name => 'oe.my_program1',
 program_type => 'PLSQL_BLOCK',
 program_action => 'BEGIN DBMS_STATS.GATHER_TABLE_STATS(''oe'', ''sales'');
END;',
 number_of_arguments => 0,
 enabled => TRUE,
 comments => 'My comments here');
END;
/
SELECT PROGRAM_NAME FROM DBA_SCHEDULER_PROGRAMS
WHERE PROGRAM_NAME = 'MY_PROGRAM1';

-- Defining Program Arguments
BEGIN
DBMS_SCHEDULER.DEFINE_PROGRAM_ARGUMENT (
 program_name => 'operations_reporting',
 argument_position => 2,
 argument_name => 'end_date',
 argument_type => 'VARCHAR2',
 default_value => '12-DEC-03');
END;
/

-- dropping a program argument
BEGIN
DBMS_SCHEDULER.DROP_PROGRAM_ARGUMENT (
 program_name => 'operations_reporting',
 argument_position => 2);
DBMS_SCHEDULER.DROP_PROGRAM_ARGUMENT (
 program_name => 'operations_reporting',
 argument_name => 'end_date');
END;
/

-- sets the value (non-Null) of an argument of the associated PROGRAM OBJECT
for a job
-- by argument position
BEGIN
DBMS_SCHEDULER.SET_JOB_ARGUMENT_VALUE(
```

```
 job_name => 'ops_reports',
 argument_position => 2,
 argument_value => '12-DEC-03');
END;
/
-- by argument name
BEGIN
DBMS_SCHEDULER.SET_JOB_ARGUMENT_VALUE(
 job_name => 'ops_reports',
 argument_name => 'END_DATE',
 argument_value => '12-DEC-03');
END;
/
-- setting an argument to NULL value
BEGIN
DBMS_SCHEDULER.SET_ATTRIBUTE_NULL (
 job_name => 'ops_reports',
 argument_name => 'END_DATE');
END;
/

-- Dropping Programs
BEGIN
 DBMS_SCHEDULER.DROP_PROGRAM('program1, program2, program3');
END;
/

-- Disabling Programs
BEGIN
DBMS_SCHEDULER.DISABLE (
 name => 'ops_reports',
 force =>FALSE, -- the default
 commit_semantics => 'STOP_ON_FIRST_ERROR'); -- TRANSACTIONAL or
ABSORB_ERRORS
END;
/

-- Enabling Programs
BEGIN
 DBMS_SCHEDULER.ENABLE('program1, program2, program3');
END;
/
```

Using Schedules

```
-- Creating Schedules
BEGIN
DBMS_SCHEDULER.CREATE_SCHEDULE (
 schedule_name => 'my_stats_schedule',
 start_date => SYSTIMESTAMP,
 end_date => SYSTIMESTAMP + INTERVAL '30' day,
 repeat_interval => 'FREQ=HOURLY; INTERVAL=4',
 comments => 'Every 4 hours');
END;
/
```

```
-- altering schedules
BEGIN
dbms_scheduler.set_attribute(
 name => 'my_stats_schedule',
 attribute => 'REPEAT_INTERVAL',
 value => 'FREQ=HOURLY; INTERVAL=2');
END;
/
attribute:
 comments
 end_date
 event_spec
 repeat_interval
 source                The host name that the database is running on
 start_date

-- Dropping Schedules
BEGIN
DBMS_SCHEDULER.DROP_SCHEDULE (
 schedule_name  => 'my_stats_schedule',
 force => FALSE); -- DEFAULT
END;
/
```

Using Job Classes

- Jobs classes provide a way to group jobs for resource allocation and prioritization, and a way to easily assign a set of attribute values to member jobs.

```
-- Creating Job Classes
BEGIN
 DBMS_SCHEDULER.CREATE_JOB_CLASS (
  job_class_name => 'finance_jobs_class',
  resource_consumer_group => 'finance_group');
END;
/

BEGIN
dbms_scheduler.set_attribute(
 name => ' finance_jobs_class',
 attribute => 'logging_level',
 value => DBMS_SCHEDULER.LOGGING_FAILED_RUNS);
END;
/
ATTRIBUTE:
 comments
 logging_level DBMS_SCHEDULER.LOGGING_OFF, LOGGING_FAILED_RUNS,
                       LOGGING_RUNS, LOGGING_FULL
 resource_consumer_group jobs run under this resource consumer group
 service

-- Dropping Job Classes
BEGIN
 DBMS_SCHEDULER.DROP_JOB_CLASS('jobclass1, jobclass2, jobclass3');
END;
/
```

Using Windows

```
-- Creating Windows
begin
 dbms_scheduler.create_window (
 window_name => 'WORK_HOURS_WINDOW',
 resource_plan => 'DAY_PLAN',
 start_date => SYSTIMESTAMP, -- also schedule_name  can be used
 repeat_interval => 'FREQ=DAILY; BYHOUR=8',
 duration => INTERVAL '10' HOUR,
 window_priority => 'HIGH'); -- or LOW
end;
/

BEGIN
DBMS_SCHEDULER.CREATE_WINDOW (
 window_name => 'my_window1',
 resource_plan => 'my_res_plan1',
 start_date => '15-JUL-03 1.00.00AM US/Pacific',
 repeat_interval => 'FREQ=DAILY',
 end_date => '15-SEP-03 1.00.00AM US/Pacific',
 duration => interval '80' MINUTE,
 comments => 'This is my first window');
END;
/

BEGIN
DBMS_SCHEDULER.CREATE_WINDOW (
 window_name => 'my_window2',
 schedule_name => 'my_stats_schedule',
 resource_plan => 'my_resourceplan1',
 duration => interval '60' minute,
 comments => 'My window');
END;
/
SELECT WINDOW_NAME FROM DBA_SCHEDULER_WINDOWS WHERE WINDOW_NAME =
'MY_WINDOW1';

-- Altering Windows
BEGIN
dbms_scheduler.set_attribute(
 name => 'MYWINDOW',
 attribute => 'window_priority',
 value => 'LOW');
END;
/
ATTRIBUTE:
 comments
 duration
 end_date
 repeat_interval
 resource_plan
 schedule_name
 start_date
 window_priority

-- openinng a window
begin
```

```
 dbms_scheduler.open_window (
 window_name => 'WORK_HOURS_WINDOW',
 duration => INTERVAL '20' MINUTE);
end;
/

-- closing a window
begin
 dbms_scheduler.close_window ('WORK_HOURS_WINDOW');
end;
/

-- dropping a window
BEGIN
 DBMS_SCHEDULER.DROP_WINDOW ('window1, window2,
 window3, windowgroup1, windowgroup2');
END;
/

-- Disabling Windows
BEGIN
DBMS_SCHEDULER.DISABLE ('sys.window1, sys.window2,
 sys.window3, sys.windowgroup1, sys.windowgroup2');
END;
/

-- Enabling Windows
BEGIN
 DBMS_SCHEDULER.ENABLE ('sys.window1, sys.window2, sys.window3');
END;
/

-- Window Logging
column log_id format a3
column window_name format a20
column operation format a8
select trim(log_id) log_id, trunc(log_date) log_date,
 window_name, operation
from dba_scheduler_window_log
/

column actual_duration format a20
select trim(log_id) log_id, trunc(log_date) log_date,
window_name, actual_duration
from dba_scheduler_window_details
order by 2 desc
/

begin
 DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE('LOG_HISTORY','60');
end;
/
```

Using Window Groups

```
If you create a window group, add windows to it, and then name this window
```

```
group in a job's schedule_name attribute, the job runs during all the windows
in the window group.
-- Creating Window Groups
BEGIN
DBMS_SCHEDULER.CREATE_WINDOW_GROUP (
 group_name => 'downtime',
 window_list => 'weeknights, weekends');
END;
/

-- Dropping Window Groups
BEGIN
 DBMS_SCHEDULER.DROP_WINDOW_GROUP('windowgroup1, windowgroup2');
END;
/

-- Adding a Member to a Window Group
BEGIN
 DBMS_SCHEDULER.ADD_WINDOW_GROUP_MEMBER ('window_group1', 'window1,
window2');
END;
/

-- Dropping a Member from a Window Group
BEGIN
 DBMS_SCHEDULER.REMOVE_WINDOW_GROUP_MEMBER('window_group1',
'window1,window2');
END;
/

-- Enabling a Window Group
BEGIN
 DBMS_SCHEDULER.ENABLE('sys.windowgroup1', 'sys.windowgroup2,
sys.windowgroup3');
END;
/

-- Disabling a Window Group: but not the members
BEGIN
 DBMS_SCHEDULER.DISABLE('sys.windowgroup1,sys.windowgroup2');
END;
/
```

Monitoring Job State with Email Notifications

- You can configure a job to send e-mail notifications when it changes state.

```
/* Configuration */
CONN / AS SYSDBA
BEGIN
  -- define the SMTP server
  DBMS_SCHEDULER.set_scheduler_attribute('email_server', 'smtp.mydomain.com:25');

  -- optionally define default sender address, which
  --  is used if the sender parameter is not specified
  DBMS_SCHEDULER.set_scheduler_attribute('email_sender', 'info@mydomain.com');
END;
/
```

```
-- to enable/disable encryption is for the SMTP server connection
--  only (11.2.0.2)
-- possible values: NONE, SSL_TLS, STARTTLS
exec DBMS_SCHEDULER.set_scheduler_attribute('email_server_encryption','SSL_TLS')


-- Authentication
-- If the SMTP server requires authentication, then the Scheduler uses the
-- user name and password stored in the specified credential object
-- default NULL
exec dbms_scheduler.create_credential('hrcredential','hr','hrpassword');
exec DBMS_SCHEDULER.set_scheduler_attribute('email_server_credential','hrcredential')



/* Using Email Notification */
-- You call ADD_JOB_EMAIL_NOTIFICATION once for each different set of notifications
-- that you want to configure for a job.

-- associate an email notification with the job
-- using the default subject and body
BEGIN
 DBMS_SCHEDULER.add_job_email_notification (
  job_name   =>  'email_notification_job',
  recipients =>  'info@ahmedbaraka.com',
  events     =>  'job_started, job_succeeded');
END;

-- subject and body specified:
BEGIN
DBMS_SCHEDULER.ADD_JOB_EMAIL_NOTIFICATION (
 job_name => 'email_notification_job',
 recipients => 'info@ahmedbaraka.com, alissa@mydomain.com',
 sender => 'do_not_reply@example.com',
 subject => 'Scheduler Job Notification-%job_owner%.%job_name%-%event_type%',
 body => '%event_type% occurred at %event_timestamp%. %error_message%',
 events => 'JOB_FAILED, JOB_BROKEN, JOB_DISABLED, JOB_SCH_LIM_REACHED');
END;


-- configures an additional e-mail notification for the same job
-- for a different event
BEGIN
DBMS_SCHEDULER.ADD_JOB_EMAIL_NOTIFICATION (
 job_name => 'email_notification_job',
 recipients => 'info@ahmedbaraka.com',
 events => 'JOB_OVER_MAX_DUR');
END;


-- The notification fires only if a job fails with "600" "700" error codes
BEGIN
 DBMS_SCHEDULER.add_job_email_notification (
   job_name         =>  'email_notification_job',
   recipients       =>  'info@ahmedbaraka.com',
   events           =>  'job_failed',
   filter_condition => ':event.error_code=600 or :event.error_code=700');
END;



/* Removing Email Notification */
```

```
-- remove the notification from specified recipient/event
BEGIN
 DBMS_SCHEDULER.remove_job_email_notification (
  job_name   =>  'email_notification_job',
  recipients =>  'info@ahmedbaraka.com',
  events     =>  'job_succeeded');
END;

-- remove the notification from all recipients and events
BEGIN
 DBMS_SCHEDULER.remove_job_email_notification (
  job_name   =>  'email_notification_job');
END;



/* Obtain Info about Email Notifications */
SELECT job_name, recipient, event, filter_condition, subject, body
FROM user_scheduler_notifications;
```

Using File Watchers

```
/* Obtain Info about FW */
SELECT file_watcher_name, destination, directory_path, file_name, credential_name
FROM   user_scheduler_file_watchers;



/* Configuration */
-- by default, a destination is checked every 10 mins. To change this:
-- only sys can do it:
CONN / AS SYSDBA
set serveroutput on
declare
 v varchar2(1000);
begin
 DBMS_SCHEDULER.GET_ATTRIBUTE ( 'FILE_WATCHER_SCHEDULE','REPEAT_INTERVAL', v);
 DBMS_OUTPUT.PUT_LINE(V);
end;
/
BEGIN
  DBMS_SCHEDULER.set_attribute(
    'file_watcher_schedule',
    'repeat_interval',
    'freq=minutely; interval=5');
END;
/




/* Creating File Watcher */
-- create OS credential:
BEGIN
  DBMS_SCHEDULER.create_credential(
    credential_name => 'fw_credential',
    username        => 'oracle',
    password        => 'oracle');
END;
/
```

```
-- Grant EXECUTE on the credential to the schema that owns the
--  event-based job that the file watcher will start:
GRANT EXECUTE ON fw_credential to DSSUSER;


-- create file watcher:
BEGIN
  DBMS_SCHEDULER.create_file_watcher(
    file_watcher_name => 'data_fw',
    directory_path    => '/tmp/test', -- if '?' = ORACLE_HOME
    file_name         => '*.dat', -- wildcard supported
    credential_name   => 'fw_credential',
    destination       => NULL, -- NULL=local server
    enabled           => FALSE);
END;
/


-- Grant EXECUTE on the file watcher to any schema that owns an event-based job
--  that references the file watcher.
GRANT EXECUTE ON data_fw to DSSUSER;


-- create a program raised by the file watcher
BEGIN
  DBMS_SCHEDULER.create_program(
    program_name        => 'import_data_prog',
    program_type        => 'stored_procedure',
    program_action      => 'import_data_proc',
    number_of_arguments => 1,
    enabled             => FALSE);
END;
/


-- define the metadata argument using the event_message attribute
--  the metadata contains info about the file, such as its name:
BEGIN
  DBMS_SCHEDULER.define_metadata_argument(
    program_name       => 'import_data_prog',
    metadata_attribute => 'event_message',
    argument_position  => 1);
END;
/


-- create the defined procedure:
-- It must accept an argument of the SCHEDULER_FILEWATCHER_RESULT type
CREATE TABLE received_files ( fileinfo VARCHAR2(4000), rdate date );

CREATE OR REPLACE PROCEDURE import_data_proc
 (p_sfwr SYS.SCHEDULER_FILEWATCHER_RESULT) AS
 v_message received_files.fileinfo%type;
BEGIN
  v_message  :=  p_sfwr.directory_path  ||  '/'  ||  p_sfwr.actual_file_name  ||  '  ('  ||
p_sfwr.file_size || ')';

  INSERT INTO received_files
  VALUES (v_message, sysdate);
  COMMIT;
END;
/
```

```
-- create the job:
BEGIN
  DBMS_SCHEDULER.create_job(
    job_name        => 'import_data_job',
    program_name    => 'import_data_prog',
    event_condition => NULL,  -- 'tab.user_data.file_size < 1024'
    queue_spec      => 'data_fw', -- file watcher name
    auto_drop       => FALSE,
    enabled         => FALSE);
END;
/



-- By default, the arrival of new files will be ignored if the job is already running.
--  If you need the job to fire for each new arrival, regardless of whether the job is
already
--  running or not, set the PARALLEL_INSTANCES attribute for the job to true. The job
--  will then be run as a lightweight job:
BEGIN
  DBMS_SCHEDULER.set_attribute('import_data_job','parallel_instances',TRUE);
END;
/



-- Enable all the objects:
EXEC DBMS_SCHEDULER.enable('data_fw','import_data_prog','import_data_job');


-- to test:
echo "This is a test" > /tmp/test/f1.dat
echo "This is a test too" > /tmp/test/f2.dat
echo "Yes another test" > /tmp/test/f3.dat


select * from received_files order by rdate desc;



/* Managing File Watchers */
-- enable/disable
EXEC DBMS_SCHEDULER.enable('data_fw') ;
EXEC DBMS_SCHEDULER.disable('data_fw') ;


-- change an attribute:
begin
 DBMS_SCHEDULER.SET_ATTRIBUTE (
  name => 'data_fw',
  attribute =>'directory_path',
  value =>'/home/oracle/receivedfiles' ) ;
end;
/

begin
 DBMS_SCHEDULER.SET_ATTRIBUTE (
  name => 'data_fw',
  attribute =>'file_name',
  value =>'*.txt' ) ;
end;
/
```

```
begin
 DBMS_SCHEDULER.SET_ATTRIBUTE (
  name => 'data_fw',
  attribute =>'credential_name',
  value =>'fw_credential2' ) ;
end;
/


-- to drop a file watchers:
DBMS_SCHEDULER.DROP_FILE_WATCHER('data_fw');
```

Using Events Raised by the Scheduler

```
 -- by default messages raised by the scheduler are
 --  deleted after 24 hours: event_expiry_time
 SET SERVEROUTPUT ON
 DECLARE
  V VARCHAR2(200);
 BEGIN
  DBMS_SCHEDULER.GET_SCHEDULER_ATTRIBUTE (
   attribute =>'EVENT_EXPIRY_TIME',
   value =>V);
  DBMS_OUTPUT.PUT_LINE(nvl(V,'24 hours')); -- 24 hours if null
 END;
 /

 declare
  n number := 48*60*60;
 begin
 DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
  attribute =>'EVENT_EXPIRY_TIME',
  value =>n); -- in seconds (24 hours if NULL)
 end;
 /

 -- making sure the required privs are granted
 -- EXECUTE on both DBMS_AQ and DBMS_AQADM
 CONN SYS
 select grantee, privilege, table_name
 from   DBA_TAB_PRIVS
 where  table_name in ('DBMS_AQ','DBMS_AQADM')
   and  grantee='USER1';

 GRANT EXECUTE ON DBMS_AQ TO USER1;
 GRANT EXECUTE ON DBMS_AQADM TO USER1;
 GRANT SELECT ON DBA_AQ_AGENTS TO USER1;
 GRANT CREATE JOB TO USER1;
 GRANT CREATE EXTERNAL JOB TO USER1;

 begin
  DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE('ENQUEUE_ANY','USER1',FALSE);
  DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE('DEQUEUE_ANY','USER1',FALSE);
  DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE('MANAGE_ANY','USER1',FALSE);
 end;
 /
```

```
-- create the job
BEGIN
 DBMS_SCHEDULER.CREATE_JOB
       (  job_name   => '"USER1"."U1JOB"',
          job_type   => 'EXECUTABLE',
          job_action => 'C:\windows\system32\cmd.exe',
         number_of_arguments => 3,
          start_date => systimestamp,
         repeat_interval => 'FREQ=SECONDLY;INTERVAL=10',
          comments   => 'Testing Events raised by scheduler',
         auto_drop  => FALSE,
         enabled    => FALSE);
 DBMS_SCHEDULER.set_job_argument_value('"USER1"."U1JOB"',1,'/c');
 DBMS_SCHEDULER.set_job_argument_value('"USER1"."U1JOB"',2,
'c:\temp\testme.bat');
 DBMS_SCHEDULER.set_job_argument_value('"USER1"."U1JOB"',3, 'passed to bat');
END;
/


-- alter job to raise events
BEGIN
 DBMS_SCHEDULER.SET_ATTRIBUTE(
   name         => '"USER1"."U1JOB"',
   attribute => 'raise_events',
   value        => dbms_scheduler.job_started +
                          dbms_scheduler.job_succeeded +
                          dbms_scheduler.job_failed +
                          dbms_scheduler.job_broken +
                          dbms_scheduler.job_completed +
                          dbms_scheduler.job_stopped +
                          dbms_scheduler.job_sch_lim_reached +
                          dbms_scheduler.job_disabled +
                          dbms_scheduler.job_chain_stalled
       );
END;
/

-- enable Job
BEGIN
   DBMS_SCHEDULER.ENABLE( '"USER1"."U1JOB"' );
END;
/


-- Consuming Scheduler-Raised Events with your Application
-- follow either Plan A or Plan B
-- Plan A
--1. conn sys or with user of MANAGE ANY QUEUE
--2. Subscribe to the queue using a new or existing agent
--3. Run the procedure DBMS_AQADM.ENABLE_DB_ACCESS(agent_name, db_username);
-- Plan B
/* DBMS_SCHEDULER.ADD_EVENT_QUEUE_SUBSCRIBER(subscriber_name);
 where subscriber_name is the name of the Oracle Streams Advanced Queuing
 (AQ) agent to be used to subscribe to the Scheduler event queue. (If it is
 NULL, an agent is created whose name is the user name of the calling user.)
This call both creates a subscription to the Scheduler event queue and grants
```

```
the user permission to dequeue using the designated agent. */
conn user1
exec DBMS_SCHEDULER.ADD_EVENT_QUEUE_SUBSCRIBER
-- opposite: REMOVE_EVENT_QUEUE_SUBSCRIBER

-- Events are dequeued from the scheduler event queue using the DBMS_AQ
SET SERVEROUTPUT ON
DECLARE
  l_dequeue_options     DBMS_AQ.dequeue_options_t;
  l_message_properties  DBMS_AQ.message_properties_t;
  l_message_handle      RAW(16);
  l_queue_msg           sys.scheduler$_event_info;
BEGIN
  l_dequeue_options.consumer_name := 'USER1';
  DBMS_AQ.dequeue(queue_name          => 'SYS.SCHEDULER$_EVENT_QUEUE',
                  dequeue_options     => l_dequeue_options,
                  message_properties  => l_message_properties,
                  payload             => l_queue_msg,
                  msgid               => l_message_handle);
  COMMIT;

  DBMS_OUTPUT.put_line ('event_type     : ' || l_queue_msg.event_type);
  DBMS_OUTPUT.put_line ('object_owner   : ' || l_queue_msg.object_owner);
  DBMS_OUTPUT.put_line ('object_name    : ' || l_queue_msg.object_name);
  DBMS_OUTPUT.put_line ('event_timestamp: ' || l_queue_msg.event_timestamp);
  DBMS_OUTPUT.put_line ('error_code     : ' || l_queue_msg.error_code);
  DBMS_OUTPUT.put_line ('event_status   : ' || l_queue_msg.event_status);
  DBMS_OUTPUT.put_line ('log_id         : ' || l_queue_msg.log_id);
  DBMS_OUTPUT.put_line ('run_count      : ' || l_queue_msg.run_count);
  DBMS_OUTPUT.put_line ('failure_count  : ' || l_queue_msg.failure_count);
  DBMS_OUTPUT.put_line ('retry_count    : ' || l_queue_msg.retry_count);
END;
/

-- remove the job and unsubscribes the user from the scheduler event queue
EXEC DBMS_SCHEDULER.drop_job('"USER1"."U1JOB"')
EXEC DBMS_SCHEDULER.remove_event_queue_subscriber
```

Using Events Raised by an Application (Events-Based Jobs)

```
-- by default messages raised by the scheduler are
--  deleted after 24 hours: event_expiry_time
declare
 n number := 48*60*60;
begin
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE (
 attribute =>'event_expiry_time',
 value =>n); -- in seconds (24 hours if NULL)
end;
/

-- making sure the required privs are granted
-- EXECUTE on both DBMS_AQ and DBMS_AQADM
select grantee, privilege, table_name
from dba_tab_privs
where table_name in ('DBMS_AQ','DBMS_AQADM')
```

```
and grantee='TEST_USER';

GRANT EXECUTE ON DBMS_AQ TO test_user;
GRANT EXECUTE ON DBMS_AQADM TO test_user;
GRANT SELECT ON DBA_AQ_AGENTS TO test_user;
begin
 DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE('ENQUEUE_ANY','test_user',FALSE);
 DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE('DEQUEUE_ANY','test_user',FALSE);
 DBMS_AQADM.GRANT_SYSTEM_PRIVILEGE('MANAGE_ANY','test_user',FALSE);
end;
/

-- CONNECT AS  test_user
-- create type for the message to receive
connect marvin/panic
create or replace type MY_MSGT as object ( msg varchar2(20) )
/

begin
 DBMS_AQADM.CREATE_QUEUE_TABLE
 ( QUEUQ_TABLE=>'MY_QT',
   QUEUQ_PAYLOAD_TYPE => 'MY_MSGT',
   MULTIPLE_CONSUMERS => TRUE );
 DBMS_AQADM.CREATE_QUEUE(
  QUEUE_NAME => 'MY_Q',
  QUEUE_TABLE => 'MY_QT' );
 DBMS_AQADM.START_QUEUE(QUEUE_NAME=>'MY_Q');
end;
/

-- example of event_condition = 'tab.user_data.event_type = ''CARD_SWIPE''
and extract hour from tab.user_data.event_timestamp < 9'
BEGIN
       dbms_scheduler.create_job
       (
          job_name => '"TEST_USER"."BCKUP_01"',
          job_type => 'EXECUTABLE',
          job_action => '/home/oracle/bin/rman.sh',
          event_condition => 'tab.user_data.msg=''GO''',
          queue_spec => '"TEST_USER"."MY_Q"', -- agent, queue name
          start_date => systimestamp,
          job_class => NULL,
          comments => 'backup a database',
         auto_drop => FALSE,
         number_of_arguments => 1,
         enable => FALSE
       );
END;
/

-- Altering a Job to Raise Events
BEGIN
 DBMS_SCHEDULER.SET_ATTRIBUTE('dw_reports', 'raise_events',
  DBMS_SCHEDULER.JOB_FAILED + DBMS_SCHEDULER.JOB_SCH_LIM_REACHED);
END;
/
raise_events    job_started CONSTANT PLS_INTEGER := 1
```

```
                job_succeeded CONSTANT PLS_INTEGER := 2
                job_failed CONSTANT PLS_INTEGER :=4
                job_broken CONSTANT PLS_INTEGER :=8
                job_completed CONSTANT PLS_INTEGER :=16
                job_stopped CONSTANT PLS_INTEGER :=32
                job_sch_lim_reached CONSTANT PLS_INTEGER :=64
                job_disabled CONSTANT PLS_INTEGER :=128
                job_chain_stalled CONSTANT PLS_INTEGER :=256
                job_all_events CONSTANT PLS_INTEGER := 511
                job_run_completed := job_succeeded + job_failed + job_stopped

-- if required, set arguments and attributes
-- job argument passed to the job
DBMS_SCHEDULER.SET_JOB_ARGUMENT_VALUE
 (
   job_name => '"TEST_USER"."BCKUP_01"',
   argument_position => 1,
   argument_value => 'db_01'
 );
-- for remote job
DBMS_SCHEDULER.SET_ATTRIBUTE
 (
    name       => '"TEST_USER"."BCKUP_01"',
    attribute => 'destination',
    value => 'pantzer:15021'
 );
DBMS_SCHEDULER.SET_ATTRIBUTE
 (
   name       => '"TEST_USER"."BCKUP_01"',
   attribute    => 'credential_name',
   value        => '"TEST_USER."JOBS_CRED2"'
 );
END;
/

-- enable Job
BEGIN
   DBMS_SCHEDULER.ENABLE( '"TEST_USER"."BCKUP_01"' );
END;
/

DECLARE
   my_msgid RAW(16);
   props dbms_aq.message_properties_t;
   enqopts dbms_aq.enqueue_options_t;
BEGIN
   sys.dbms_aq.enqueue('marvin.bckup_q', enqopts, props,
                     marvin.bckup_msgt('GO'), my_msgid);
end;
/
COMMIT;

-- * alternatively scheduler can be used for the same
BEGIN
DBMS_SCHEDULER.CREATE_EVENT_SCHEDULE (
 schedule_name => 'entry_events_schedule',
 start_date => SYSTIMESTAMP,
```

```
 event_condition => 'tab.user_data.event_type = ''CARD_SWIPE''',
 queue_spec => 'entry_events_q, entry_agent1');
END;
/

BEGIN
 DBMS_SCHEDULER.SET_ATTRIBUTE ('entry_events_schedule', 'event_spec',
'tab.user_data.event_type = ''BAD_BADGE''', 'entry_events_q, entry_agent1');
END;
/
```

Using Chains

```
-- Creating Chains
BEGIN
 DBMS_SCHEDULER.CREATE_CHAIN (
  chain_name => 'my_chain1',
  rule_set_name => NULL,
  evaluation_interval => NULL,
  comments => 'My first chain');
END;
/

-- Defining Chain Steps
--  a step point to program, another chain, event
BEGIN
 DBMS_SCHEDULER.DEFINE_CHAIN_STEP (
  chain_name => 'my_chain1',
  step_name => 'my_step1',
  program_name => 'my_program1');

 DBMS_SCHEDULER.DEFINE_CHAIN_STEP (
  chain_name => 'my_chain1',
  step_name => 'my_step2',
  program_name => 'my_chain2');
END;
/

-- event-based (waits for an event to occur)
BEGIN
 DBMS_SCHEDULER.DEFINE_CHAIN_EVENT_STEP (
  chain_name => 'my_chain1',
  step_name => 'my_step3',
  event_schedule_name => 'my_event_schedule');
END;
/

-- Adding Rules to a Chain
--  Each rule has a condition and an action
--  rule that starts the chain always evaluates to TRUE
--  A chain job does not complete until one of the rules containing the END
action
--  evaluates to TRUE. If no such a step, the job enters the CHAIN_STALLED
state.
BEGIN
```

```
 DBMS_SCHEDULER.DEFINE_CHAIN_RULE (
  chain_name => 'my_chain1',
  condition => 'TRUE',
  action => 'START step1',
  rule_name => 'my_rule1',
  comments => 'start the chain');
 DBMS_SCHEDULER.DEFINE_CHAIN_RULE (
  chain_name => 'my_chain1',
  condition => 'step1 completed',
  action => 'START step2',
  rule_name => 'my_rule2');
END;
/
/*
 :step_name.attribute. (step_name refers to a typed object.) Possible
attributes are:
 completed, state, start_date, end_date, error_code,
 and duration. Possible values for the state attribute
 include: 'NOT_STARTED', 'SCHEDULED', 'RUNNING',
 'PAUSED', 'STALLED', 'SUCCEEDED', 'FAILED', and
 'STOPPED'. If a step is in the state 'SUCCEEDED', 'FAILED',
 or 'STOPPED', its completed attribute is set to 'TRUE',
 otherwise completed is 'FALSE'.
*/
-- examples of conditions
step3 succeeded
step3 failed

-- Example: 2
BEGIN
DBMS_SCHEDULER.CREATE_CHAIN (
 chain_name => 'my_chain1',
 rule_set_name => NULL,
 evaluation_interval => NULL,
 comments => NULL);
END;
/
--- define three steps for this chain.
BEGIN
 DBMS_SCHEDULER.DEFINE_CHAIN_STEP('my_chain1', 'step1', 'my_program1');
 DBMS_SCHEDULER.DEFINE_CHAIN_STEP('my_chain1', 'step2', 'my_program2');
 DBMS_SCHEDULER.DEFINE_CHAIN_STEP('my_chain1', 'step3', 'my_program3');
END;
/
--- define corresponding rules for the chain.
BEGIN
DBMS_SCHEDULER.DEFINE_CHAIN_RULE('my_chain1', 'TRUE', 'START step1');
DBMS_SCHEDULER.DEFINE_CHAIN_RULE (
'my_chain1', 'step1 COMPLETED', 'Start step2, step3');
DBMS_SCHEDULER.DEFINE_CHAIN_RULE (
'my_chain1', 'step2 COMPLETED AND step3 COMPLETED', 'END');
END;
/

-- Example 3:
BEGIN
DBMS_SCHEDULER.CREATE_CHAIN (
```

```
 chain_name => 'my_chain2',
 rule_set_name => NULL,
 evaluation_interval => NULL,
 comments => NULL);
END;
/
--- define three steps for this chain.
BEGIN
 DBMS_SCHEDULER.DEFINE_CHAIN_STEP('my_chain2', 'step1', 'my_program1');
 DBMS_SCHEDULER.DEFINE_CHAIN_STEP('my_chain2', 'step2', 'my_program2');
 DBMS_SCHEDULER.DEFINE_CHAIN_STEP('my_chain2', 'step3', 'my_program3');
END;
/
--- define corresponding rules for the chain.
BEGIN
 DBMS_SCHEDULER.DEFINE_CHAIN_RULE ('my_chain2', 'TRUE', 'START step1');
 DBMS_SCHEDULER.DEFINE_CHAIN_RULE ( 'my_chain2', 'step1 SUCCEEDED', 'Start
step2');
 DBMS_SCHEDULER.DEFINE_CHAIN_RULE ('my_chain2', 'step1 COMPLETED AND step1
NOT SUCCEEDED', 'Start step3');
 DBMS_SCHEDULER.DEFINE_CHAIN_RULE ('my_chain2', 'step2 COMPLETED OR step3
COMPLETED', 'END');
END;
/

-- Enabling Chains
BEGIN
 DBMS_SCHEDULER.ENABLE ('my_chain1');
END;
/

-- Creating Jobs for Chains
-- either use the RUN_CHAIN procedure or create a job of type
--  'CHAIN'. job action refers to chain name
--  a step job created for every step
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
 job_name => 'chain_job_1',
 job_type => 'CHAIN',
 job_action => 'my_chain1',
 repeat_interval => 'freq=daily;byhour=13;byminute=0;bysecond=0',
 enabled => TRUE);
END;
/

-- Dropping Chains
BEGIN
 DBMS_SCHEDULER.DROP_CHAIN (
 chain_name => 'my_chain1',
 force => TRUE);
END;
/

-- Running Chains
BEGIN
DBMS_SCHEDULER.RUN_CHAIN (
 chain_name => 'my_chain1',
 job_name => 'quick_chain_job',
```

```
 start_steps => 'my_step1, my_step2');
END;
/

-- Dropping Rules from a Chain
BEGIN
 DBMS_SCHEDULER.DROP_CHAIN_RULE (
  chain_name => 'my_chain1',
  rule_name => 'my_rule1',
  force => TRUE);
END;
/

-- Disabling Chains
BEGIN
 DBMS_SCHEDULER.DISABLE ('my_chain1');
END;
/

-- Dropping Chain Steps
BEGIN
 DBMS_SCHEDULER.DROP_CHAIN_STEP (
  chain_name => 'my_chain2',
  step_name => 'my_step2',
  force => TRUE);
END;
/

-- Altering Chain Steps
BEGIN
DBMS_SCHEDULE.ALTER_CHAIN (
 chain_name => 'my_chain1',
 step_name => 'my_step3',
 attribute => 'SKIP',
 value => TRUE);
END;
/
-- for a running step
BEGIN
DBMS_SCHEDULER.ALTER_RUNNING_CHAIN (
 job_name => 'my_job1',
 step_name => 'my_step1',
 attribute => 'PAUSE',
 value => TRUE);
END;
/

-- Handling Stalled Chains
-- option1: alter the state of one of its steps with the
-- check ALL_SCHEDULER_RUNNING_CHAINS and ALL_SCHEDULER_CHAIN_RULES
BEGIN
DBMS_SCHEDULER.ALTER_RUNNING_CHAIN (
 job_name => 'my_job1',
 step_name => 'my_step1',
 attribute => 'SUCCEEDED',
 value => TRUE);
END;
```

```
/
-- option2:
-- if one or more rules are incorrect, you can use the
--  DEFINE_CHAIN_RULE procedure to replace them
--  check ALL_SCHEDULER_CHAIN_RULES
--  After adding or updating rules, you must run EVALUATE_RUNNING_CHAIN
```

Allocating Resources Among Jobs

- For jobs, resource allocation is specified by associating a job class with a consumer group, or by associating a job class with a database service name and mapping that database service to a consumer group.

```
BEGIN
 DBMS_SCHEDULER.CREATE_JOB_CLASS (
  job_class_name => 'finance_jobs',
  resource_consumer_group => 'finance_group');
END;
/
```

Administering Oracle Scheduler

Configuring Oracle Scheduler

```
-- Setting Scheduler Privileges
GRANT SCHEDULER_ADMIN TO username;
-- create jobs, schedules, or programs
GRANT CREATE JOB TO scott;
-- alter, or drop windows, job classes, or window groups
GRANT MANAGE SCHEDULER TO adam;

-- Setting Chain Privileges
-- create a chain in his own schema, create rules, rule sets,
--  and evaluation contexts in his own schema
BEGIN
 DBMS_RULE_ADM.GRANT_SYSTEM_PRIVILEGE(DBMS_RULE_ADM.CREATE_RULE_OBJ,
'username');
 DBMS_RULE_ADM.GRANT_SYSTEM_PRIVILEGE (DBMS_RULE_ADM.CREATE_RULE_SET_OBJ,
'username');
 DBMS_RULE_ADM.GRANT_SYSTEM_PRIVILEGE
(DBMS_RULE_ADM.CREATE_EVALUATION_CONTEXT_OBJ, 'username');
END;
/

-- Setting Scheduler Attributes
-- default_timezone
select dbms_scheduler.stime from dual;
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE('default_timezone','US/Eastern');
SELECT DISTINCT TZNAME FROM V$TIMEZONE_NAMES;
-- log_history
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE('log_history','90');
-- max_job_slave_processes (default NULL)
-- event_expiry_time
SET SERVEROUTPUT ON
DECLARE
```

```
 V VARCHAR2(200);
BEGIN
 DBMS_SCHEDULER.GET_SCHEDULER_ATTRIBUTE (
  attribute =>'EVENT_EXPIRY_TIME',
  value =>V);
 DBMS_OUTPUT.PUT_LINE(nvl(V,'24 hours')); -- 24 hours if null
END;
/
declare
 n number := 48*60*60;
begin
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
 attribute =>'EVENT_EXPIRY_TIME',
 value =>n); -- in seconds (24 hours if NULL)
end;
/
```

Monitoring and Managing the Scheduler

```
-- Viewing the Currently Active Window and Resource Plan
SELECT WINDOW_NAME, RESOURCE_PLAN
FROM DBA_SCHEDULER_WINDOWS
WHERE ACTIVE='TRUE';


-- Finding Information About Currently Running Jobs
SELECT JOB_NAME, STATE FROM DBA_SCHEDULER_JOBS
WHERE JOB_NAME = 'MY_EMP_JOB1';

SELECT JOB_NAME , STATE, START_DATE, END_DATE, LAST_START_DATE,
NEXT_RUN_DATE, LAST_RUN_DURATION, FAILURE_COUNT
FROM USER_SCHEDULER_JOBS
ORDER BY  START_DATE DESC;

-- progress of currently running jobs
SELECT * FROM ALL_SCHEDULER_RUNNING_JOBS;

-- job that is part of a running chain
SELECT * FROM ALL_SCHEDULER_RUNNING_CHAINS
WHERE JOB_NAME='MY_JOB1';


-- Monitoring and Managing Window and Job Logs
SELECT JOB_NAME, OPERATION, OWNER FROM DBA_SCHEDULER_JOB_LOG;


-- Job Run History Details
select log_id,
 job_name, status,
 to_char(log_date, 'DD-MON-YYYY HH24:MI') log_date
 from dba_scheduler_job_run_details
where job_name = 'MY_JOB14';


-- Controlling Job Logging: at job or class levels
logging_level attribute in the CREATE_JOB_CLASS:
DBMS_SCHEDULER.LOGGING_OFF, DBMS_SCHEDULER.LOGGING_RUNS,
DBMS_SCHEDULER.LOGGING_FULL
```

```
DBMS_SCHEDULER.SET_ATTRIBUTE (
'mytestjob', 'logging_level', DBMS_SCHEDULER.LOGGING_FULL);


-- Window Logs
SELECT LOG_ID, TO_CHAR(LOG_DATE, 'MM/DD/YYYY'), WINDOW_NAME, OPERATION
FROM DBA_SCHEDULER_WINDOW_LOG;
SELECT LOG_ID, WINDOW_NAME, ACTUAL_START_DATE, ACTUAL_DURATION
FROM DBA_SCHEDULER_WINDOW_DETAILS;


-- Purging Logs
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE('log_history','90'); -- days
 -- can be specified at job class create
DBMS_SCHEDULER.SET_ATTRIBUTE('class2','log_history','30');
-- Purging Logs Manually
DBMS_SCHEDULER.PURGE_LOG();
 -- urges all entries from the jog log that are
 --  older than three days. The window log is not affected
DBMS_SCHEDULER.PURGE_LOG(log_history => 3, which_log => 'JOB_LOG');
DBMS_SCHEDULER.PURGE_LOG(log_history => 10, job_name => 'job1, sys.class2');


-- Changing Job Priorities
-- 1(high)-5
BEGIN
 DBMS_SCHEDULER.SET_ATTRIBUTE (
  name => 'my_emp_job1',
  attribute => 'job_priority',
  value => 1);
END;
/
SELECT JOB_NAME, JOB_PRIORITY FROM DBA_SCHEDULER_JOBS;


-- Monitoring Running Chains
SELECT *
FROM USER_SCHEDULER_RUNNING_CHAINS
WHERE JOB_NAME = 'MY_CHAIN_JOB';
```

Enabling, Using and Disabling Remote External Jobs

```
-- 1) Setting Up the Database
conn sys
-- verify that the XML DB option is installed:
 DESC RESOURCE_VIEW
-- Enable HTTP connections to the database:
-- port between 1 and 65536, and for UNIX and Linux greater than 1023
BEGIN
 DBMS_XDB.SETHTTPPORT(port);
END;
/
-- run script
SQL> @?/rdbms/admin/prvtrsch.plb
-- Set a registration password
BEGIN
 DBMS_SCHEDULER.SET_AGENT_REGISTRATION_PASS('mypassword');
END;
```

```
/

-- 2) Installing, Configuring, and Starting the Scheduler Agent
-- install Scheduler agent from Oracle Database Gateways in
--  the Database Media Pack in its own HOME
-- To install, configure, and start the Scheduler agent on a remote host:
-- Log in to the remote host as oracle
-- Run the Oracle Universal Installer
-- -> Next
-- -> select Oracle Scheduler Agent -> Next
-- -> Home
-- -> eneter hostname, port
-- -> install
-- -> run script_path/root.sh as root
-- -> Exit
-- cehck the agent configuration parameters in the file: schagent.conf

-- Register the Scheduler agent with a database that is to run remote
external jobs on
--  the agent's host computer. Use the following command:
AGENT_HOME/bin/schagent -registerdatabase db_host db_http_port
where:
 db_host is the host name or IP address of the host on which the database
resides.
 db_http_port is the port number that the database listens on for HTTP
connections.
SELECT DBMS_XDB.GETHTTPPORT() FROM DUAL; -- zero means disabled

Repeat the previous step for each database that is to run remote external
jobs on the agent's host.

Start the Scheduler agent with the following command:
AGENT_HOME/bin/schagent -start

-- 3) Stopping the Scheduler Agent
On UNIX and Linux:
AGENT_HOME/bin/schagent -stop
On Windows, stop the service OracleSchedulerExecutionAgent

-- 4) Disabling Remote External Jobs
DROP USER REMOTE_SCHEDULER_AGENT CASCADE;
Registration of new scheduler agents and execution of remote external jobs is
disabled until you run prvtrsch.plb again.



/* to create a remote external job */
-- create a credential object
exec dbms_scheduler.create_credential('hrcredential','hr','hrpassword');

-- Grant privileges
grant execute on system.hrcrdential to someuser;

-- create the remote external job:
begin
 dbms_scheduler.create_job(
  job_name => 'remove_logs',
  job_type => 'executable',
```

```
   job_action => '/u01/app/oracle/logs/removelogs',
   repeat_interval => 'freq=daily; byhour=23',
   enabled => false);
end;
/

-- set the CREDENTIAL_NAME attribute
exec
dbms_scheduler.set_attribute('remove_logs','credential_name','hrcredential');

-- set the DESTINATION attribute
exec dbms_scheduler.set_attribute('remove_logs','destination',
'localhost.localdomain:1521');

--enable the external job
exec dbms_scheduler.enable('remove_logs');
```

Import/Export and the Scheduler

- You must use the Data Pump utilities (impdp and expdp) to export Scheduler objects.
- After you import Scheduler credentials, you must reset the passwords using the SET_ATTRIBUTE procedure of the DBMS_SCHEDULER package.

Scheduler Privileges

- System Privileges:
  - CREATE JOB
  - CREATE ANY JOB
  - CREATE EXTERNAL JOB
  - EXECUTE ANY PROGRAM
  - EXECUTE ANY CLASS
  - MANAGE SCHEDULER
- Object Privileges: on jobs, programs, chains, schedules and job classes.
  - EXECUTE
  - ALTER
  - ALL

Scheduler Data Dictionary Views

- DBA_SCHEDULER_CHAINS
- DBA_SCHEDULER_CHAIN_RULES
- DBA_SCHEDULER_CHAIN_STEPS
- DBA_SCHEDULER_CREDENTIALS
- DBA_SCHEDULER_GLOBAL_ATTRIBUTE
- DBA_SCHEDULER_JOBS
- DBA_SCHEDULER_JOB_ARGS
- DBA_SCHEDULER_JOB_CLASSES
- DBA_SCHEDULER_JOB_LOG
- DBA_SCHEDULER_JOB_ROLES
- DBA_SCHEDULER_JOB_RUN_DETAILS
- DBA_SCHEDULER_PROGRAMS
- DBA_SCHEDULER_PROGRAM_ARGS
- DBA_SCHEDULER_REMOTE_DATABASES

- DBA_SCHEDULER_REMOTE_JOBSTATE
- DBA_SCHEDULER_RUNNING_CHAINS
- DBA_SCHEDULER_SCHEDULES
- DBA_SCHEDULER_WINDOWS
- DBA_SCHEDULER_WINDOW_DETAILS
- DBA_SCHEDULER_WINDOW_GROUPS
- DBA_SCHEDULER_WINDOW_LOG
- DBA_SCHEDULER_WINGROUP_MEMBERS

Using the UTL_FILE Package

```
CREATE DIRECTORY MYDIR AS '/home/oracle/';
GRANT READ, WRITE ON DIRECTORY utl_dir to .. | public;

DECLARE
 fHandle UTL_FILE.FILE_TYPE;
 v_username dba_users.username%TYPE;
 CURSOR users IS SELECT username FROM dba_users order by username;
 n number := 0;
BEGIN
 -- options: r w a
 fHandle := UTL_FILE.FOPEN('MYDIR','utlfile.txt','w');
 UTL_FILE.PUT_LINE(fHandle,'SN'||CHR(9) || 'User NAME');
 UTL_FILE.FCLOSE(fHandle);

 /* re-Open the utlfile.txt for append, and get its file handle */
 fHandle := UTL_FILE.FOPEN('MYDIR','utlfile.txt','a');
OPEN users;
LOOP
 FETCH users INTO v_username;
 EXIT when users%NOTFOUND;
 n := n + 1;
 /* Write a line of text to the file utlfile.txt */
 UTL_FILE.PUT_LINE(fHandle,n || chr(9) || v_username);
 /* Read a line from the file utltext.txt */
 -- UTL_FILE.GET_LINE(fHandle,v_username||v_failed||v_life||v_lock);
END LOOP;
CLOSE users;
UTL_FILE.FCLOSE(fHandle);
EXCEPTION
WHEN UTL_FILE.INVALID_PATH THEN
 RAISE_APPLICATION_ERROR(-20100,'Invalid Path');
WHEN UTL_FILE.INVALID_MODE THEN
 RAISE_APPLICATION_ERROR(-20101,'Invalid Mode');
WHEN UTL_FILE.INVALID_OPERATION then
 RAISE_APPLICATION_ERROR(-20102,'Invalid Operation');
WHEN UTL_FILE.INVALID_FILEHANDLE then
 RAISE_APPLICATION_ERROR(-20103,'Invalid Filehandle');
WHEN UTL_FILE.WRITE_ERROR then
 RAISE_APPLICATION_ERROR(-20104,'Write Error');
WHEN UTL_FILE.READ_ERROR then
 RAISE_APPLICATION_ERROR(-20105,'Read Error');
WHEN UTL_FILE.INTERNAL_ERROR then
 RAISE_APPLICATION_ERROR(-20106,'Internal Error');
WHEN OTHERS THEN
 UTL_FILE.FCLOSE(fHandle);
```

```
END;
/
```

Data Loading and Transforming Tools

Oracle's ETL (Extraction-Transformation-Loading) solution includes the following components:

- SQL*Loader: see the section Using SQL*Loader Utility in Oracle Database Utilities part.
- External tables: see Managing External Tables
- Multitable inserts
- Merging Data
- Table functions: Table functions produce a set of rows as output. Instead of defining the transform declaratively in SQL, you define it procedurally in PL/SQL. See Table Functions.
- Transportable tablespaces: see Transporting Tablespaces Between Databases
- Oracle Warehouse Builder (OWB): OWB offers you a wizard-driven facility to load data into the database through SQL*Loader, load data from an Oracle database or other databases such as Sybase, Informix, and Microsoft SQL Server via Oracle Transparent Gateways.

Using Database Links

```
/* Info about DB Links */
select OWNER, DB_LINK, USERNAME, HOST, CREATED from DBA_DB_LINKS;


/* Privs */
grant CREATE DATABASE LINK to hr;

grant CREATE PUBLIC DATABASE LINK to hr;


/* Private */
CONNECT system/system_passwd@mydb

CREATE DATABASE LINK MONITOR
CONNECT TO hr IDENTIFIED BY hr
USING 'monitor';

Create database link orcl2.net connect to sa identified by a
using '(DESCRIPTION=(ADDRESS = (PROTOCOL = TCP)(HOST =10.4.x.x) (PORT=1521))
(connect_data=(service_name=orcl)))';

/* Public */
CREATE PUBLIC DATABASE LINK MONITOR
 CONNECT TO hr IDENTIFIED BY hr USING 'monitor';
```

Managing Diagnostic Data

Setting the Automatic Diagnostic Repository Directory

- It replaces USER_DUMP_DEST, BACKGROUND_DUMP_DEST and CORE_DUMP_DEST parameters.
- Default value: $ORACLE_BASE/diag/rdbms/$INSTANCE_NAME/$ORACLE_SID
- If you haven't set the ORACLE_BASE variable, the value of the DIAGNOSTIC_DEST parameter defaults to $ORACLE_HOME/log.

```
show parameter DIAGNOSTIC_DEST

SELECT VALUE FROM V$PARAMETER WHERE NAME ='diagnostic_dest';

ALTER SYSTEM SET DIAGNOSTIC_DEST ='C:\ORACLE\diag';

SELECT NAME, VALUE FROM V$DIAG_INFO;
```

Using adrci Tool

General usage of adrci

```
adrci –help
adrci>help

adrci>help show incident

-- running adrci in batch mode
adrci exec 'command [; comamnd]. . . '
adrci script=file_name


adrci>show base

adrci>show homes
adrci>show homepath
adrci>set homepath diag\rdbms\ora11g\ora11g


-- spooling
adrci>spool /u01/myfiles/myadrci.txt
adrci> ...
adrci>spool off


-- view alert log
set editor notepad.exe
show alert
```

```
show alert –tail 30
show alert –p "MESSAGE TEXT LIKE '%ORA-600%'*


-- List Trace Files
show tracefile


-- View Incidents
show incident
show incident –mode detail –p "incident_id=112564"
```

Using adrci to Package Incidents

With `adrci` tool, you can package all the diagnostic files related to specific problems into a ZIP file to submit it to Oracle support. To do so, you use special commands called IPS as shown in the following steps:

1. Create a logical package: use `ips create package` command to create an empty logical package as shown in the example below. The package will be given a serially generated number.

   ```
   adrci>ips create package
   ```

2. Add diagnostic data to the logical package: this is done by `ips add incident` command as shown below:

   ```
   adrci>ips add incident 112564 package 1
   ```

   Actually, there are formats of the `ips create package` command which enables you to perform the steps 1 and 2 in one command. Following are those command formats:

   - `ips create package problem`
   - `ips create package problem key`
   - `ips create package incident`
   - `ips create package time`

3. Generate the physical package. The files related to the incident will be collected in a ZIP file. The following example shows the command to perform this task:

   ```
   adrci>ips generate package 1 in /u01/myfiles/incidents
   ```

   If you decide to add or change any diagnostic data later, you can do so by generating an *incremental* ZIP file. Modify the command as follows to achieve that:

   ```
   adrci>ips generate package 1 in /u01/myfiles/incidents incremental
   ```

   You will notice that the generated file has the phase INC in its name indicating that it is an incremental ZIP file.

   `ips` commands behavior is controlled by various configuration options. To display those configuration options, use the command `ips show configuration`.

Managing Database Health Monitor

To display list of the check that can be performed, issue the following query:

```
SELECT NAME, DESCRIPTION, OFFLINE_CAPABLE FROM V$HM_CHECK;
```

The OFFLINE_CAPABLE column defines whether you can perform the check when the database is offline or not.

Running Health Checks Using the DBMS_HM

A DBA can use DBMS_HM to manually invoke the database check. To retrieve the list of checks that can be run manually by users, issue the following query:

```
SELECT NAME FROM V$HM_CHECK WHERE INTERNAL_CHECK = 'N';
```

Use the procedure RUN_CHECK to perform a database health check. Its first parameter CHECKNAME is mandatory and it takes one of the returned names by the query above.

```
exec DBMS_HM.RUN_CHECK(CHECK_NAME=>'DB Structure Integrity Check', RUN_NAME=>'HM01');
```

Most health checks accept input parameters. You can view parameter names and descriptions with the V$HM_CHECK_PARAM view. Some parameters are mandatory while others are optional. The following query displays parameter information for all health checks:

```
select C.NAME CHECK_NAME, P.NAME PARAMETER_NAME, P.TYPE,
       P.DEFAULT_VALUE, P.DESCRIPTION
from   V$HM_CHECK_PARAM P, V$HM_CHECK C
where  P.CHECK_ID = C.ID and C.INTERNAL_CHECK = 'N'
order by C.NAME;
```

Input parameters are passed to the INPUT_PARAMS argument of the RUN_CHECK procedure as name/value pairs separated by semicolons (;). The following example illustrates how to pass the transaction ID as a parameter to the Transaction Integrity Check:

```
begin
 DBMS_HM.RUN_CHECK (
  CHECK_NAME  =>  ' Transaction  Integrity  Check',          --  passed  value  is  case
sensitive
  RUN_NAME => 'MY_RUN', INPUT_PARAMS => 'TXN_ID=7.33.2');
end;
```

Database Health checks executions are stored in ADR and can be viewed by either querying the V$HM_RUN:

```
SELECT * FROM V$HM_RUN;
```

Another option is to run the adrci command show hm_run:

```
adrci>show hm_run
```

You can view a report of a particular Health check by using the following adrci command:

```
adrci>show report hm_run HM01
```

Alternatively, you can DBMS_HM package as shown in the following code example:

```
declare
 v_rpt clob;
begin
 v_rpt := DBMS_HM.GET_RUN_REPORT('HM01');
end;
```

Findings, if any, detected by the checks can be obtained from V$HM_FINDING and recommendations from V$HM_RECOMMENDATION.

Running Health Checks Using the Enterprise Manager

After connecting as SYSDBA, under the **Advisor Central** page, you will see **Checkers** link which can be used to manually invoke any Health check.

Managing Data Recovery Advisor

Data Recovery Advisor is an Oracle Database 11g tool that automatically diagnoses data failures, determines and presents appropriate repair options, and executes repairs at the user's request. Data Recovery Advisor can diagnose failures such as the following:

- Inaccessible components like datafiles and control files.
- Physical corruptions such as block checksum failures and invalid block header
- Field values
- Inconsistent datafiles (online and offline)
- I/O failures

The advisor however doe not recover from failures on standby databases or RAC environment. This advisor can be used through RMAN or the Enterprise Manager.

Using Data Recovery Advisor with RMAN

Following are the RMAN commands to use Data Recovery Advisor:

1. List failures by running the `LIST FAILURE` command. Following are variations of using the command:

```
RMAN>LIST FAILURE;
RMAN>LIST OPEN;
RMAN>LIST CLOSED;
```

2. Optionally, execute `LIST FAILURE ... DETAIL` to list details of an individual failure.

```
RMAN>LIST FAILURE 105 DETAIL;
```

3. If you suspect that failures exist that have not been automatically diagnosed by the database, then run `VALIDATE DATABASE` to check for corrupt blocks and missing files. If a failure is detected, then RMAN logs it into the ADR, where it can be accessed by the Data Recovery Advisor.

4. Determine repair options by running the `ADVISE FAILURE` command.

```
RMAN>ADVISE FAILURE;
```

5. Choose a repair option. You can repair the failures manually or run the `REPAIR FAILURE` command to fix them automatically. By default, the `REPAIR FAILURE` command prompts the user to confirm the repair, but this can be prevented using the `NOPROMPT` keyword. Be aware that the previous command must be issued before using `REPAIR FAILURE` command.

   The following form of the command informs you how RMAN plans to repair the failure:

```
RMAN>REPAIR FAILURE PREVIEW
```

6. You may wish to change the priority of a failure (to `HIGH` or `LOW`), if it does not represent a problem to you, or even manually close it. This can be done by the `CHANGE FAILURE` command:

```
RMAN> CHANGE FAILURE 202 PRIORITY LOW;
```

Note Data Recovery Advisor may detect or handle some logical corruptions. But in general, corruptions of this type require help from Oracle Support Services.

Using Data Recovery Advisor with the Enterprise Manager

Access the Data Recovery Advisor in the Enterprise Manager by following the links: **Availability> Manage> Perform Recovery> Perform Automated Repair**

Using SQL Test Case Builder

The SQL Test Case Builder aims at capturing the information pertaining to a SQL-related problem, along with the exact environment under which the problem occurred, so that the problem can be reproduced

and tested on a separate Oracle database instance. Once the test case is ready, you can upload the problem to Oracle Support to enable support personnel to reproduce and troubleshoot the problem.

The information gathered by SQL Test Case Builder includes the query being executed, table and index definitions (but not the actual data), PL/SQL functions, procedures, and packages, optimizer statistics, and initialization parameter settings.

The output of the SQL Test Case Builder is a SQL script that contains the commands required to recreate all the necessary objects and the environment.

Accessing SQL Test Case Builder Using DBMS_SQLDIAG

The DBMS_SQLDIAG has a procedure named EXPORT_SQL_TESTCASE which is used to generate a SQL test case for a given SQL statement, SQL Id (taken from V$SQL) or an incident id. Following steps should be followed:

1. Create directory to hold the SQL test case files.

```
CREATE DIRECTORY sql_tes_dir AS 'C:\Oracle\TestCase';
```

2. Execute the proper form of EXPORT_SQL_TESTCASE procedure. Following is an example using a passed SQL statement.

```
DECLARE
 V_SQL CLOB := 'SELECT * FROM HR.NAMES WHERE ID BETWEEN 100 AND 1000';
 V_TESTCASE CLOB;
BEGIN
 DBMS_SQLDIAG.EXPORT_SQL_TESTCASE (
  DIRECTORY      =>'SQL_TES_DIR',
  SQL_TEXT       =>V_SQL,
  USER_NAME      =>'HR',
  BIND_LIST      =>NULL,
  EXPORTENVIRONMENT   =>TRUE,
  EXPORTMETADATA      =>TRUE,
  EXPORTDATA          =>TRUE,
  SAMPLINGPERCENT     =>100,
  CTRLOPTIONS         =>NULL,
  TIMELIMIT           =>0,
  TESTCASE_NAME       =>'RETURN_NAMES', -- generated scripts prefix
  TESTCASE            =>V_TESTCASE);
END;
```

Accessing SQL Test Case Builder Using the Enterprise Manager

From Enterprise Manager, the SQL Test Case Builder is accessible only when a SQL incident occurs. SQL-related problem is referred to as a SQL incident.

To access the SQL Test Case Builder, follow the links **the Support Workbench page> Click on an incident ID> Investigate and Resolve section> Oracle Support> Generate Additional Dumps and Test Cases> click on the icon in the Go To Task**.

Patching Oracle Products

Using Oracle Opatch

References

Oracle Universal Installer and OPatch User's Guide
On metalink: OPatch Utility Guide - 10.2 [ID 554417.1]

How to Download It

On metalink, search Patch 6880880.
At document writing, it is on:
https://updates.oracle.com/ARULink/PatchDetails/process_form?patch_num=6880880

Environment Variables OPatch Uses

ORACLE_HOME - Oracle home location.
OPATCH_DEBUG - Log level that specifies the amount of logging OPatch should perform.
OPATCH_PLATFORM_ID - Unique platform ID.
PATH - Path information.

Backup Recommendations

It is highly recommended that you back up the ORACLE_HOME before any patch operation. You can use any method, such as zip, cp -r, tar, and cpio.

Info about a Utility

To find out more about the options available with each command, please use the following syntax:

```
opatch util <Utility_Name> -help
```

lsinventory

It is used to check what is currently installed on the system.

```
opatch lsinventory -detail
to list all available  Oracle homes:
opatch lsinventory -all
```

NApply

It is used to apply a set of patches under a directory.
opatch napply <patch_location> -id 1,2,3 -skip_subset -skip_duplicate

This will apply patches 1, 2, and 3 which are under < the patch_location> directory. OPatch will skip duplicate patches and subset patches (patches under <patch_location> that are subsets of patches installed in the ORACLE_HOME)

NRollback

It is used to rollback a set of patches installed in an ORACLE_HOME. You can invoke the command using "opatch nrollback" or "opatch util nrollback".

opatch nrollback -id 1,2,3

UpdateRemoteNodes (RAC)

It is used to propagate/remove files/directories to/from remote nodes using files under ORACLE_HOME/.patch_storage/<ID>/rac/*.

The directories listed in copy_dirs.txt will be copied to remote nodes.
The files listed in copy_files.txt wil be copied to remote nodes.
The directories listed in remove_dirs.txt will be deleted from remote nodes.

The files listed in remove_files.txt will be deleted from remote nodes.

Cleanup

It is used to clean up 'restore.sh, make.txt' files and 'rac, scratch, backup' directories in the ORACLE_HOME/.patch_storage directory. If -ps option is used, then it cleans the above specified areas only for that patch, else for all patches under ORACLE_HOME/.patch_storage. You will be still able to rollback patches after this cleanup.

opatch util cleanup -ps 6121193_Jun_21_2008_04_19_82

CopyListedFiles (RAC)

It is used to copy all files listed in ORACLE_HOME/.patch_storage/<ID>/rac/copy_files.txt to remote nodes. If -fp option is used, then one can specify the path of the file containing the list of files to be copied. The files mentioned in this file will be copied to the remote nodes.

opatch util copylistedfiles -fp a -remote_nodes ceintcb-a5

CopyListedFilesTest (RAC)

It is used to copy a single file to remote nodes. The usage remains the same as CopyListedFiles.

opatch util copylistedfilestest -fp /home/oracle/a -remote_nodes ceintcb-a5

CopyListedDirs (RAC)

It is used to recursively copy all directories listed in ORACLE_HOME/.patch_storage/<ID>/rac/copy_dirs.txt to remote nodes. If -dp option is used, then one can specify the path of the file containing the list of directories to be copied. The directories mentioned in this file will be copied to the remote nodes.

CopyListedDirsTest (RAC)

It is used to copy a single file to remote nodes. The usage remains the same as CopyListedDirs.

RemoveListedFiles (RAC)

It is used to remove files listed in ORACLE_HOME/.patch_storage/<ID>/rac/remove_files.txt on remote nodes. If -fr option is used, then one can specify the path of the file containing the list of files to be removed. The files mentioned in this file will be removed from the remote nodes.

RemoveListedFilesTest (RAC)

It is used to remove a single file from remote nodes. The usage remains the same as RemoveListedFiles.

RemoveListedDirs (RAC)

It is used to recursively remove directories listed in ORACLE_HOME/.patch_storage/<ID>/rac/remove_dirs.txt from remote nodes. If -dr option is used, then one can specify the path of the file containing the list of directories to be removed. The directories mentioned in this file will be removed from the remote nodes.

RemoveListedDirsTest (RAC)

It is used to remove a single directory from remote nodes. The usage remains the same as RemoveListedDirs.

RunLocalMake

It is used to invoke re-link on the local node. The make commands are stored in ORACLE_HOME/.patch_storage/<ID>/make.txt. You need to use the -ps option to specify the Patch ID with timestamp. A directory by this name will be present under ORACLE_HOME/.patch_storage. The make.txt file present under ORACLE_HOME/.patch_storage/Patch ID with timestamp/ will be used to perform the local make operation. This command cannot be run if you have already run Cleanup as it would have removed these make.txt files.

opatch util runlocalmake -ps 6121250_Jun_21_2007_04_16_11

RunRemoteMake (RAC)

It is used to invoke re-link on remote nodes. The make commands are stored in ORACLE_HOME/.patch_storage/<ID>/rac/makes_cmd.txt. The usage remains the same as RunLocalMake.

RunAnyCommand (RAC)

It is used to run any command on remote nodes. The command should be specified using the -cmd option.

opatch util runanycommand -remote_nodes ceintcb-a5 -cmd ls

Verify

It is used to run the patch verification process to ensure that the patch was applied to the ORACLE_HOME. It uses the defined ORACLE_HOME and the given patch location via -ph, to run the check.

opatch util verify -ph ~/6646853/6121183

LoadXML

It is used to check the validity of an XML file. The -xmlInput option can be used to specify the path of the xml file.

opatch util loadxml -xmlInput $ORACLE_HOME/inventory/ContentsXML/comps.xml

Part 2 Oracle Database Net Services

Connectivity Naming Methods

The Local Naming Method

- The TNS_ADMIN environment variable tells Oracle where to locate tnsnames.ora and sqlnet.ora files.

```
/* Starting Oracle Net Configuration Assistant */
export DISPLAY=172.16.14.15:0.0
netca
```

The Easy Connect Naming Method

- The EZCONNECT keyword should be in the NAMES.DIRECTORY_PATH variable in the sqlnet.ora file
- You can't use any advanced features of Oracle networking such as connection pooling, external procedure calls, or Heterogeneous Services.

```
sqlplus system/system_passwd@myhost.myorg.org:1521/mydb.myorg.org
# the default port number is 1521
sqlplus system/system_passwd@myhost.myorg.org/mydb.myorg.org
```

The External Naming Method

The external naming method uses external naming services such as the Network Information Service (NIS), originally developed by Sun Microsystems, to resolve net service names.

1. Have your system administrator configure NIS if it isn't already in place.
2. Create a tnsnames.ora file as you would in the local naming method.
3. Convert the tnsnames.ora file to a tnsnames map, which you'll need for the NIS server later

on. You can derive the tnsnames map from the tnsnames.ora file by having your system administrator use the tns2nis command, as shown here:

```
# tns2nis tnsnames.ora
```

4. Copy the tnsnames map file to the server on which the NIS is running.
5. Install the tnsnames map file on the NIS server using the makedbm NIS program, as shown

here:

```
# makedbm tnsnames /var/yp/'domainname'/tnsnames
```

6. Test the NIS installation of the tnsnames map by using the following command:

```
# ypmatch net_service_name tnsnames
```

You should get a confirmation back in the following form:

```
description=(address=(protocol=tcp)
(host=host_name)(port=port_number)))
(connect_data=(service_name=service_name)))
```

7. Edit the sqlnet.ora file as follows:

```
NAMES_DIRECTORY_PATH=(nis, hostname, tnsnames)
```

The nis method should be listed first inside the brackets so that Oracle Net will attempt to resolve the service name using NIS first. Apart from that, the order of the items in the brackets doesn't matter.

The Directory Naming Method

The directory naming method stores database connection information in a Lightweight Directory Access Protocol (LDAP)–compliant directory server (like Oracle Internet Directory). The connect identifiers are stored under an Oracle context that contains entries for use with OID.

Database Resident Connection Pooling (DRCP)

- DRCP (11g) is especially designed to help architectures such as PHP with the Apache server, that can't take advantage of middle-tier connection pooling because they used multiprocess single-threaded application servers. DRCP enables applications such as these to easily scale up to server connections in the tens of thousands.
- DRCP is controlled by the following configuration parameters:

INACTIVITY_TIMEOUT	*maximum idle time for a pooled server before it is terminated.*
MAX_LIFETIME_SESSION	time to live TTL duration for a pooled session.
MAX_USE_SESSION	maximum number of times a connection can be taken and released to the pool.
MAX_SIZE and MIN_SZIE	the maximum and minimum number of pooled servers in the connections pool.
INCRSIZE	pool would increment by this number of pooled server when pooled server are unavailable at application request time.
MAX_THINK_TIME	maximum time of inactivity by the client after getting a server from the pool. If the client does not issue a database call after grabbing a server from the pool, the client will be forced to relinquish control of the pooled server and will get an error. The freed up server may or may not be returned to the pool.
SESSION_CACHED_CURSORS	turn on SESSION_CACHED_CURSORS for all connections in the pool. This is an existing initialization parameter

```
/* Enabling and Disabling DRCP */
conn sys as sysdba
-- the ramins open after DB restart
exec dbms_connection_pool.start_pool();
select connection_pool, status, maxsize from dba_cpool_info;
exec dbms_connection_pool.stop_pool();

-- specify using DRCP
-- in EZCONNECT method (.Net 11g)
myhost.comany.com:1521/mydb.company.com:POOLED
-- tnsnames
mydb = (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=myhost.company.com)
(SERVER=POOLED)))
```

```
/* Configuring DRCP */
begin
 DBMS_CONNECTION_POOL.ALTER_PARAM( PARAM_NAME =>'INACTIVITY_TIMEOUT',
PARAM_VALUE=>'3600');
end;
/
-- restore parameter values to their defaults
exec dbms_connection_pool.restore_defaults()

/* Monitor DRCP */
SELECT
STATUS,MINSIZE,MAXSIZE,INCRSIZE,SESSION_CACHED_CURSORS,INACTIVITY_TIMEOUT
FROM DBA_CPOOL_INFO;

SELECT NUM_OPEN_SERVERS, NUM_BUSY_SERVERS, NUM_REQUESTS, NUM_HITS
       NUM_MISSES, NUM_WAITS, NUM_PURGED, HISTORIC_MAX
FROM V$CPOOL_STATS;
-- class-level stats
Select * From V$CPOOL_CC_STATS
```

Oracle and Java Database Connectivity

Establishing Database Connectivity

- JDBC driver types:
  - JDBC thin driver
  - JDBC OCI driver
  - JDBC server-side thin driver
  - JDBC server-side internal driver

```
-- load the drivers
-- method 1
DriverManager.registerDriver ("new oracle.jdbc.OracleDriver()");
-- method 2
Class.forName("oracle.jdbc.driver.OracleDriver");

-- establish the connection
connection conn=DriverManager.getConnection(
"jdbc:oracle:thin:@prod1:1521:finprod", username, passwd);

-- Creating the Statement Object
statement stmt = conn.createStatement();

-- Handling Queries
string first_name,last_name;
number salary;
resultSet rs = stmt.executeQuery("SELECT * FROM Employees");
while (rs.next()) {
 first_name = rs.getString("first_name");
 last_name = rs.getString("last_name");
 salary = rs.getNumber("salary");
 system.out.println(first_name + last_name +" with salary of:" + salary);
 }

-- Handling DDL and Nonquery DML Statements
statement stmt = conn.createStatement();
 stmt.executeUpdate("CREATE TABLE Emp" + "(last_name VARCHAR2(30), first_name
VARCHAR2(20), salary number");
-- auto committed by default
stmt.executeUpdate("INSERT INTO Emp " +
 "VALUES ('Lname', 'Fname', salary)");

-- transaction control
conn.setAutoCommit(false);
conn.commit();
conn.rollback();

-- Error Handling
```

```
try { conn.setAutoCommit(false);
  stmt.executeUpdate("..");
  conn.commit();
  conn.setAutoCommit(true);
}
catch(SQLException ex) {
 system.err.println("SQLException: " + ex.getMessage());
 conn.rollback();
 conn.setAutoCommit(true);
}
```

Miscellaneous Connectivity Options

Setting the Default Connect String

```
-- in unix
export TWO_TASK=mydb
-- in windows: set LOCAL key to mydb value in the registry or:
SET LOCAL=<mydb>

-- later you don't have to specify the connect string
sqlplus scott/tiger
sqlplus [ [<option>] [<logon>] [<start>] ]
```

Installing the Instant Client

- Oracle's new Instant Client software allows you to run your applications without installing the standard Oracle Client or having an ORACLE_HOME.

```
(1) Download and install Oracle Instant Client software. You must install the
Basic client package and tehn you can also include any of the advanced
optional packages.
http://www.oracle.com/technology/software/tech/oci/instantclient/index.html
(2) Unzip the downloaded package and copy it wherever you like.
(3) In Linux, set LD_LIBRARY_PATH to instantclient directory. On Windows, set
PATH to instantclient
```

Setting Listener Options

```
/* Queue Size */
-- concurrent connection requests can be made
LISTENER=
(DESCRIPTION=
 (ADDRESS=(PROTOCOL=tcp)(HOST=myserver)(PORT=1521)(QUEUESIZE=10)))

/* Password */
-- ask for a password before executing any admin command
LSNRCTL> set password
LSNRCTL> change_password
```

Setting Access Controls

```
-- in the sqlnet.ora
-- only the addresses in the list are allowed to make connections
tcp.validnode_checking = yes
tcp.invited_nodes = (server1.us.mycompany.com,172.14.16.152)
-- addresses in the list are excluded
```

```
tcp.excluded_nodes = (server1.us.mycompany.com,172.14.16.152)
```

Changing Windows Hostname

Step 1 – Create Hosts Entry for Old Hostname

Locate your hosts file, typically located at %WINDIR%\system32\drivers\etc\hosts and add an entry for the old (current) hostname.

```
#
# HOSTS file
#

win2k3r2    172.16.10.10
```

Note the IP address – this is the address of a Loopback Adapter installed on the guest machine. As outlined by the Oracle Installer, a Loopback Adapter is required on systems that do not have a static IP address (as do virtual machines using NAT, etc.)

Step 2 – Uninstall Enterprise Manager Console

Because there are configuration settings stored with Enterprise Manager Console that reference the hostname, the same must be uninstalled.

```
emca -deconfig dbcontrol db -repos drop
```

Note, before executing this command, ensure that the Oracle instance is running – it has to be in order for Enterprise Manager Configuration Assistance to drop the repository and de-configure the Console.

Step 3 – Stop All Oracle Services

Once the uninstall of Enterprise Manage Console has completed, stop all Oracle Services on the guest machine.

- iSQL*Plus Service – typically named Oracle<OracleHomeName>iSQL*Plus
- Oracle Listener Service – typically named Oracle<OracleHomeName>TNSListener
- Oracle Database Instance Service – typically named OracleServer<SID>

Step 4 – Update listener.ora and tnsnames.ora

Once all the Oracle services have stopped, update the listener.ora and tnsnames.ora files, located in %ORACLE_HOME%\network\admin to reflect the desired (new) hostname.

```
LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
(ADDRESS = (PROTOCOL = TCP)(HOST = win2k3r2)(PORT = 1521))
)
)
```

```
DEVBOX =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = win2k3r2)(PORT = 1521))
(CONNECT_DATA =
```

```
(SERVER = DEDICATED)
(SERVICE_NAME = devbox)
)
)
```

Step 5 – Rename Host and Restart

Now, rename the computer and restart the guest machine.

Step 6 – Ensure Oracle Instance is Running

Once the guest machine has started up, log in and ensure the Oracle instance is running using the following command line (typically required, unless the instance, not the Windows Service, is configured to auto-start.)

```
oradim -startup –sid devbox
```

Step 7 – Reinstall Enterprise Manager Console

After ensuring the Oracle instance is running, reinstall Enterprise Manager Console using the following command line:

```
emca -config dbcontrol db -repos create
```

Step 8 – Validate Enterprise Manager Console Installation

Lastly, after the successful installation of Enterprise Manager Console, validate the installation by navigating to the logon page – typically http://<hostname>:1158/em/.

Part 3 Oracle Database Backup and Recovery

Backup Guidelines

Causes of Unplanned Down Time

- Software Failures
  - Operating system
  - Database
  - Middleware
  - Application
  - Network
- Hardware Failures
  - CPU
  - Memory
  - Power supply
  - Bus
  - Disk
  - Tape
  - Controllers
  - Network
  - Power
- Human Errors
  - Operator error
  - User error
  - DBA
  - System admin.
  - Sabotage
- Disasters
  - Fire
  - Flood
  - Earthquake
  - Power failure
  - Bombing

Causes of Planned Down Time

- Routine Operations
  - Backups
  - Performance mgmt
  - Security mgmt
  - Batches
- Periodic Maintenance
  - Storage maintenance
  - Initialization parameters

  - Software patches
  - Schema management
  - Operating system
  - Middleware
  - Network
- New deployments
  - HW upgrade
  - OS upgrades
  - DB upgrades
  - MidW upgrades
  - App upgrades
  - Net upgrades

Oracle's Solution to Down Time

Failure Category	Failure Sub-Category	Solutions
Unplanned Down Time	System Failures	RAC Data Guard Streams Fast-start Fault Recovery RAC Data Guard Streams
	Data Failures	RMAN backup/recovery ASM Flashback Hardware Assisted Resilient Data (HARD) Data Guard and Streams
Planned Down Time	System Changes	Rolling upgrades Dynamic provisioning
	Data Changes	Online redefinition

Minimizing Unplanned Downtime Guidelines

- Use RAID data storage with mirroring (1+0 is a good choice).
- Maintain offsite storage of your backups with a reliable vendor. Make is part of your recovery testing program.
- Normally, for a production database, operating in Archivelog mode is a must.
- Multiplex the control files on separate disk drives managed by different disk controllers.
- Oracle strongly recommends that you multiplex the redo log file.
- After every major structural change, back up the control file. You can take backup of the control file every hour.
- If you backup to a tape, backup to two copies. The media might be defective.
- Make auxiliary files part of your backup: (SPFILE) or the init.ora, sqlnet.ora, tnsnames.ora, password and wallet files.
- Log your backup operations.
- Make every application has its own tablespace.
- Use Data Pump utility for supplemental protection.
- Make a plan to make sure that the backups are actually readable and valid.

- Make database recovery testing plan.
- Always keep a redundancy set online (use flash recovery area for this purpose) so you can recover faster. A redundancy set has:
  - Last backup of all datafiles
  - Last backup of the control file
  - Multiplexed copies of the current redo log files
  - Copies of the current control file
  - The archived redo logs since the last backup
  - Auxilary files: SPFILE or the init.ora, listener.ora, and tnsnames.ora, pwsd

SLA Sample

Standard Processing Services. The Provider shall furnish and allow access to the processing environments

listed below:

a. Mid-tier processing.

(1) Applications to be processed:

Financial Information Systems (FIS) to include:

LIST OF FIS APPLICATIONS

Other Departmental Applications

(2) Hours of Availability.

Interactive: Monday-Friday* 07:00-17:00*

Saturday, Sunday and Holidays Not Applicable

*Application will be a web-based 24×7×365 system WITH the exception of the scheduled

maintenance periods (see below)

Batch: Not applicable

Maintenance: Monthly, Fourth Weekend of Every Month

(3) Standard Processing/Service Requirements.

All of the systems/applications listed in paragraph (1) above are required to be operational 98% of the total time listed in paragraph (2) above. The Information Systems Department will provide a method for the Department of Finance to monitor operational percentages.

(4) Processing of data will be limited to the functionality/processing that was being conducted at the time of handing over the operations to the Information Services Department.

...

List Database failure possible reasons or scenarios and the time required to recover for each reason.

Planning a Backup Strategy Guidelines

- If possible, have your required backup files on disks (faster than tapes).
- Automatically delete obsolete backups.
- If DML load makes it feasible, use incremental backups.

Examples of Backup Schedules for a Database

```
/* Example 1 */
# you'll have: image copy of the database (L0),
#   incremental backup L1
#   Archived redo logs between current time and L0
RUN {
   # apply L1 to the specified datafile images
   RECOVER COPY OF DATABASE WITH TAG 'incr_update';
   # create L0 datafile images (first time) then L1
   BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'incr_update'
DATABASE;
  }


/* Example 2 */
# you'll have image copy of db L0, 3 L1s,
#  archived log between current time and L0
# assumption: Size the flash recovery area so it holds three days' worth of
incremental backups.
RUN
{
 RECOVER COPY OF DATABASE TAG "whole_db_copy" UNTIL TIME 'SYSDATE-3';
 BACKUP INCREMENTAL LEVEL 1
 FOR RECOVER OF COPY WITH TAG "whole_db_copy" DATABASE;
}
```

User-Managed Backups

Obtaining Database File Information

```
SELECT name, status FROM v$datafile;

SELECT t.name tablespace, f.name datafile
FROM v$tablespace t, v$datafile f
WHERE t.ts# = f.ts#
ORDER BY t.name;

SELECT name FROM v$controlfile;

select member from V$LOGFILE
```

Making Whole Closed Database Backups

```
# 1) shutdown the db
# 2) copy all db physical files: datafiles, control files, redo log files,
parameter file, password file, wallet file (if there's a one).
Ensure that the complete pathnames of the files are noted

#!/bin/ksh
ORACLE_SID=$1
export ORACLE_SID
export ORAENV_ASK=NO
BACKUP_DIR=/u02/app/oracle
. oraenv
sqlplus -s system/mypsswrd << EOF
SET HEAD OFF FEED OFF ECHO OFF TRIMSPOOL ON LINESIZE 200
SPOOL /u01/app/oracle/dba/cold_backup.ksh
SELECT 'cp ' ||file_name|| ' ${BACKUP_DIR}' from sys.dba_data_files;
SELECT 'cp ' ||name || ' ${BACKUP_DIR}' from V$controlfile;
SELECT 'cp ' ||member|| ' ${BACKUP_DIR}' from V$logfile;
SPOOL OFF;
EXIT;
EOF

# 3) open the db
```

Making a Whole Open Backup

```
#!/bin/ksh
ORACLE_SID=$1
export ORACLE_SID
export ORACLE_ASK=NO
BACKUP_DIR=/u01/app/oracle/backup
export BACKUP_DIR
sqlplus -s "sys/sys_password as sysdba" << EOF
set linesize 200
set head off
set feed off
SPOOL /u01/app/oracle/dba/hot_backup.ksh
BEGIN
dbms_output.put_line ('alter database begin backup;');
for f1 in (select file_name fn from sys.dba_data_files)
```

```
loop
 dbms_output.put_line( 'host cp '||f1.fn|| ' $BACKUP_DIR');
end loop;
dbms_output.put_line ('alter database end backup;');
dbms_output.put_line('alter database backup controlfile to '|| '
$BACKUP_DIR/control'|| ';');
dbms_output.put_line('alter system switch logfile;');
END;
/
SPOOL OFF;
EXIT
EOF
```

Making Tablespace Backups

```
# offline
ALTER TABLESPACE users OFFLINE;
copy datafiles
ALTER TABLESPACE users ONLINE;

# online
ALTER TABLESPACE sysaux BEGIN BACKUP;
copy datafiles
ALTER TABLESPACE sysaux END BACKUP;
```

Obtaining Backup Status Information

```
SELECT * FROM v$backup;
```

Checking Datafiles Taken as Backup

- The utility checks only for logical corruption below the HWM.

```
dbv file=D:\ORACLE\ORADATA\ORA10G\USERS01.DBF  blocksize=4096
dbv file=D:\ORACLE\ORADATA\ORA10G\USERS01.DBF  blocksize=8192
```

Handling Crash Before User-Manged Backup Ends

```
#1) make sure that one or more datafiles left in online backup mode:
SELECT * FROM v$backup;

#2)
ALTER DATABASE END BACKUP;

#3)
SELECT * FROM v$backup;

#4)
ALTER DATABASE OPEN;
```

Backing up Control File

```
# create a text trace file:
ALTER DATABASE BACKUP CONTROLFILE TO TRACE

# binary copy of the control file
ALTER DATABASE BACKUP CONTROLFILE to 'c:\temp\controlfile.ctl'
```

Backing Up Initialization Files

```
# create pfile
CREATE SPFILE='/u01/oracle/dbs/test_spfile.ora' FROM
PFILE='/u01/oracle/dbs/test_init.ora'

# In Oracle 11g, write current values of instance parameters
CREATE PFILE FROM MEMORY;
CREATE SPFILE FROM MEMORY;
```

User-Managed Complete Recovery

User-Managed Recovery in NOARCHIVELOG Mode

```
#(1
SHUTDOWN ABORT

#(2
copy datafiles, control files and redo log files from backup location into the
original destinations

#(3
STARTUP
```

User-Managed Recovery in NOARCHIVELOG Mode Without Redo Log File

```
#(1
SHUTDOWN IMMEDIATE

#(2
Restore the most recent whole database backup with operating system commands.

#(3 mimic incomplete recovery:
SQL>RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE;
SQL>CANCEL

#4)
SQL> ALTER DATABASE OPEN RESETLOGS;
```

User-Managed Complete Recovery in ARCHIVELOG Mode

- If the lost file is the SYSTEM, UNDO or SYSAUX, the databse will be mostly closed.

```
# Checking steps:
#Identifying datafiles that need recovery
SELECT * FROM v$recover_file;
SELECT file_id f#, file_name,
 tablespace_name tablespace, status
 FROM dba_data_files;
# Locating Archived Log Files to Apply
SELECT * FROM v$recovery_log;

#1) if the database is closed, open mount
STARTUP MOUNT

#2) make sure the datafile is offline
select file#, name , status from v$datafile;
ALTER DATABASE datafile '/disk2/data/df2.dbf' offline;

#3) restore the damaged or lost datafile using os commands
if it should be resotred in a new location:
```

```
 ALTER DATABASE RENAME FILE '/ORADATA/u03/users01.dbf'
  TO '/ORADATA/u04/users01.dbf';

#4) if archive redo log file are stored in location different from
log_archive_dest_n:
#4.1) Specifying the location and name at the recover prompt:
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
#4.2)Using the ALTER SYSTEM ARCHIVE command:
SQL> ALTER SYSTEM ARCHIVE LOG START TO <new location>;
#4.3)Using the RECOVER FROM <LOCATION> command:
SQL> RECOVER FROM '<new location>' DATABASE
#4.4)
SQL>SET LOGSOURCE /new_directory
#4.5)
ALTER DATABASE RECOVER FROM '/new_directory';

#5) recover the datafile(s)
RECOVER [AUTOMATIC] DATABASE
RECOVER [AUTOMATIC] TABLESPACE <NUMBER> | <NAME>
RECOVER [AUTOMATIC] DATAFILE <'filename'> | <NAME>

#6) if db is open, take the datafile online:
ALTER DATABASE DATAFILE '/disk2/data/df2.dbf' ONLINE;
ALTER TABLESPACE user_data ONLINE;

#7) if required:
ALTER DATABASE OPEN;
```

Re-Creating Lost Datafiles Without Backup

```
# if the database open:
ALTER TABLESPACE table_data OFFLINE IMMEDIATE;
# if the database is closed, open mount
STARTUP MOUNT
ALTER DATABASE DATAFILE 4 OFFLINE;

# re-create the datafile with the same name:
ALTER DATABASE CREATE DATAFILE '/ORADATA/u03/users01.dbf';
#or
#re-create the datafile but with a new filename or location:
ALTER DATABASE CREATE DATAFILE '/ORADATA/u03/users01.dbf' AS
'/ORADATA/u04/users01.dbf';

RECOVER TABLESPACE table_data;
RECOVER DATAFILE '/ORADATA/u03/users01.dbf ';

ALTER TABLESPACE table_data ONLINE;
ALTER DATABASE DATAFILE '/ORADATA/u03/users01.dbf' ONLINE;
```

User-Managed Incomplete Recovery

Common Situations Requiring Incomplete Recovery

- Missing archive
- Loss of redo logs
- User error
- Loss of control files: but you have a backup of an old binary copy

User-Managed Incomplete Recovery Steps

You must have the following to recover:

- A valid offline or online backup containing all datafiles.
- All archived redo logs, from the restored backup to before the time of failure.

```
1. Perform a full closed backup of the existing database.

2. Restore all datafiles. You may need to restore archived logs.
If archive log changed to different destination:
SET LOGSOURCE <LOCATION>
STARTUP MOUNT

3. Place the database in mount mode and insure that the datafiles are online.
SELECT file_id f#, file_name,
 tablespace_name tablespace, status
 FROM dba_data_files;

4. Recover the database.
RECOVER [AUTOMATIC] DATABASE until ..
 until time 'YYYY-MM-DD:HH:MI:SS'
 until cancel
 until scn <integer>
 using backup control file

recover database until time '2002-03-09:11:44:00'


5. Open the database by using the RESETLOGS option and verify the recovery.
alter database open resetlogs;

Note: If log files need to be re-created on another disk due to media failure,
use the ALTER DATABASE DROP LOG GROUP and ALTER DATABASE ADD LOG GROUP
commands to create the log files manually.

6. Perform a whole closed backup of the database.
```

Recovering from Lost Control File by Re-Creating the Control File

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
the command generates file in
<ORACLE_ADR>\rdbms\<GSID>\<SID>\trace\<SID>_ora_nnn.trc
The exact file name will also be found in
<ORACLE_ADR>\rdbms\<GSID>\<SID>\trace\alert_<SID>.log
```

```
-- The following are current System-scope REDO Log Archival related
-- parameters and can be included in the database initialization file.
--
-- LOG_ARCHIVE_DEST=''
-- LOG_ARCHIVE_DUPLEX_DEST=''
--
-- LOG_ARCHIVE_FORMAT=ARC%S_%R.%T
--
-- DB_UNIQUE_NAME="ora11g"
--
-- LOG_ARCHIVE_CONFIG='SEND, RECEIVE, NODG_CONFIG'
-- LOG_ARCHIVE_MAX_PROCESSES=4
-- STANDBY_FILE_MANAGEMENT=MANUAL
-- STANDBY_ARCHIVE_DEST=%ORACLE_HOME%\RDBMS
-- FAL_CLIENT=''
-- FAL_SERVER=''
--
-- LOG_ARCHIVE_DEST_10='LOCATION=USE_DB_RECOVERY_FILE_DEST'
-- LOG_ARCHIVE_DEST_10='OPTIONAL REOPEN=300 NODELAY'
-- LOG_ARCHIVE_DEST_10='ARCH NOAFFIRM NOEXPEDITE NOVERIFY SYNC'
-- LOG_ARCHIVE_DEST_10='REGISTER NOALTERNATE NODEPENDENCY'
-- LOG_ARCHIVE_DEST_10='NOMAX_FAILURE NOQUOTA_SIZE NOQUOTA_USED
NODB_UNIQUE_NAME'
-- LOG_ARCHIVE_DEST_10='VALID_FOR=(PRIMARY_ROLE,ONLINE_LOGFILES)'
-- LOG_ARCHIVE_DEST_STATE_10=ENABLE
--
-- LOG_ARCHIVE_DEST_1='LOCATION=C:\oracle\oracledb11g\RDBMS'
-- LOG_ARCHIVE_DEST_1='MANDATORY NOREOPEN NODELAY'
-- LOG_ARCHIVE_DEST_1='ARCH NOAFFIRM EXPEDITE NOVERIFY SYNC'
-- LOG_ARCHIVE_DEST_1='NOREGISTER NOALTERNATE NODEPENDENCY'
-- LOG_ARCHIVE_DEST_1='NOMAX_FAILURE NOQUOTA_SIZE NOQUOTA_USED
NODB_UNIQUE_NAME'
-- LOG_ARCHIVE_DEST_1='VALID_FOR=(PRIMARY_ROLE,ONLINE_LOGFILES)'
-- LOG_ARCHIVE_DEST_STATE_1=ENABLE
--
-- Below are two sets of SQL statements, each of which creates a new
-- control file and uses it to open the database. The first set opens
-- the database with the NORESETLOGS option and should be used only if
-- the current versions of all online logs are available. The second
-- set opens the database with the RESETLOGS option and should be used
-- if online logs are unavailable.
-- The appropriate set of statements can be copied from the trace into
-- a script file, edited as necessary, and executed when there is a
-- need to re-create the control file.
--
--     Set #1. NORESETLOGS case
--
-- The following commands will create a new control file and use it
-- to open the database.
-- Data used by Recovery Manager will be lost.
-- Additional logs may be required for media recovery of offline
-- Use this only if the current versions of all online logs are
-- available.
-- After mounting the created controlfile, the following SQL
-- statement will place the database in the appropriate
-- protection mode:
--  ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE
```

```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "ORA11G" NORESETLOGS  NOARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 'C:\ORACLE\ORADATA\ORA11G\REDO01.LOG'  SIZE 50M,
  GROUP 2 'C:\ORACLE\ORADATA\ORA11G\REDO02.LOG'  SIZE 50M,
  GROUP 3 'C:\ORACLE\ORADATA\ORA11G\REDO03.LOG'  SIZE 50M
-- STANDBY LOGFILE
DATAFILE
  'C:\ORACLE\ORADATA\ORA11G\SYSTEM01.DBF',
  'C:\ORACLE\ORADATA\ORA11G\SYSAUX01.DBF',
  'C:\ORACLE\ORADATA\ORA11G\UNDOTBS01.DBF',
  'C:\ORACLE\ORADATA\ORA11G\USERS01.DBF',
  'C:\ORACLE\ORADATA\ORA11G\FDA_TBS.DBF'
CHARACTER SET AR8MSWIN1256
;
-- Commands to re-create incarnation table
-- Below log names MUST be changed to existing filenames on
-- disk. Any one log file from each branch can be used to
-- re-create incarnation records.
-- ALTER DATABASE REGISTER LOGFILE
'C:\ORACLE\FLASH_RECOVERY_AREA\ORA11G\ARCHIVELOG\2010_03_19\O1_MF_1_1_%U_.ARC'
;
-- Recovery is required if any of the datafiles are restored backups,
-- or if the last shutdown was not normal or immediate.
RECOVER DATABASE
-- Database can now be opened normally.
ALTER DATABASE OPEN;
-- Commands to add tempfiles to temporary tablespaces.
-- Online tempfiles have complete space information.
-- Other tempfiles may require adjustment.
ALTER TABLESPACE TEMP ADD TEMPFILE 'C:\ORACLE\ORADATA\ORA11G\TEMP01.DBF'
     SIZE 20971520  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE 32767M;
-- End of tempfile additions.
--
--     Set #2. RESETLOGS case
--
-- The following commands will create a new control file and use it
-- to open the database.
-- Data used by Recovery Manager will be lost.
-- The contents of online logs will be lost and all backups will
-- be invalidated. Use this only if online logs are damaged.
-- After mounting the created controlfile, the following SQL
-- statement will place the database in the appropriate
-- protection mode:
--  ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE PERFORMANCE
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "ORA11G" RESETLOGS  NOARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
```

```
LOGFILE
  GROUP 1 'C:\ORACLE\ORADATA\ORA11G\REDO01.LOG'  SIZE 50M,
  GROUP 2 'C:\ORACLE\ORADATA\ORA11G\REDO02.LOG'  SIZE 50M,
  GROUP 3 'C:\ORACLE\ORADATA\ORA11G\REDO03.LOG'  SIZE 50M
-- STANDBY LOGFILE
DATAFILE
  'C:\ORACLE\ORADATA\ORA11G\SYSTEM01.DBF',
  'C:\ORACLE\ORADATA\ORA11G\SYSAUX01.DBF',
  'C:\ORACLE\ORADATA\ORA11G\UNDOTBS01.DBF',
  'C:\ORACLE\ORADATA\ORA11G\USERS01.DBF',
  'C:\ORACLE\ORADATA\ORA11G\FDA_TBS.DBF'
CHARACTER SET AR8MSWIN1256
;
-- Commands to re-create incarnation table
-- Below log names MUST be changed to existing filenames on
-- disk. Any one log file from each branch can be used to
-- re-create incarnation records.
-- ALTER DATABASE REGISTER LOGFILE
'C:\ORACLE\FLASH_RECOVERY_AREA\ORA11G\ARCHIVELOG\2010_03_19\O1_MF_1_1_%U_.ARC'
;
-- Recovery is required if any of the datafiles are restored backups,
-- or if the last shutdown was not normal or immediate.
RECOVER DATABASE USING BACKUP CONTROLFILE
-- Database can now be opened zeroing the online logs.
ALTER DATABASE OPEN RESETLOGS;
-- Commands to add tempfiles to temporary tablespaces.
-- Online tempfiles have complete space information.
-- Other tempfiles may require adjustment.
ALTER TABLESPACE TEMP ADD TEMPFILE 'C:\ORACLE\ORADATA\ORA11G\TEMP01.DBF'
     SIZE 20971520  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE 32767M;
-- End of tempfile additions.
--
```

Flash Recovery Area

Obtaining Information on Flash Recovery Area

```
-- FRA Configuration
show parameter DB_RECOVERY_FILE_DEST_SIZE
show parameter DB_RECOVERY_FILE_DEST
-- related parameters
show parameter DB_CREATE_FILE_DEST
show parameter LOG_ARCHIVE_DEST_1
show parameter LOG_ARCHIVE_DEST_2

-- space usage
select NAME, round(SPACE_LIMIT/1024/1024/1024,4) SPACE_LIMIT_GB,
 round(SPACE_USED/1024/1024/1024,4) SPACE_USED_GB,
 round(SPACE_RECLAIMABLE/1024/1024/1024,4)  SPACE_RECLAIMABLE_GB,
 NUMBER_OF_FILES
from V$RECOVERY_FILE_DEST;

select FILE_TYPE, PERCENT_SPACE_USED, PERCENT_SPACE_RECLAIMABLE,
NUMBER_OF_FILES
from V$FLASH_RECOVERY_AREA_USAGE
order by PERCENT_SPACE_USED desc
```

Configuring Flash Recovery Area

- If configured, LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST parameters cannot be used.

```
-- configuring FRA
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE = 4G SCOPE=BOTH;
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST = 'C:\ORACLE\RECOVERY_AREA' SCOPE=BOTH
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST = '+dskgrp1'

-- related parameters
DB_CREATE_FILE_DEST = /u02/test/oradata/dbfiles/
LOG_ARCHIVE_DEST_1 = 'LOCATION=/u03/test/arc_dest1'
LOG_ARCHIVE_DEST_2 = 'LOCATION=USE_DB_RECOVERY_FILE_DEST'

-- disable FRA
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST = ''

-- 85% warning and 79% critical threshold are by default configured
```

Backing Up the Flash Recovery Area

- In order to back up the flash recovery area itself using RMAN, you must set CONFIGURE BACKUP OPTIMIZATION to ON.
- Neither of the two commands, BACKUP RECOVERY AREA nor BACKUP RECOVERY FILES, will back up any permanent files or the flashback logs in the flash recovery area
- You can back up the flash recovery area only to a tape device using the following backup commands:

```
BACKUP RECOVERY AREA
```

- This command backs up all flash recovery files in the current or previous flash recovery area destinations.
- It backs up only those files that have never been backed up to tape before.
- The files that the command will back up include full backups, incremental backups, control file autobackups, archive logs, and datafile copies.

`BACKUP RECOVERY FILES`

This command backs up all the files that the `BACKUP RECOVERY AREA` command does, but from all areas on your file system, not just from the flash recovery area.

`BACKUP RECOVERY FILE DESTINATION`

Use this command to move disk backups created in the flash recovery area to tape.

Moving the Flash Recovery Area

```
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='/u01/app/oracle/new_area' SCOPE=BOTH
```

- Eventually, Oracle will delete all the transient files from the previous flash recovery area location, when each of them becomes eligible for deletion. However, if you want to move your current permanent files, transient files, or flashback logs to the new flash recovery area, you can do so by using the standard file-moving procedures.
- If the database flashback logs are enabled and you moved the flash recovery area, the flashback option should be switched off and on.

Recovery Manager (RMAN)

Using A Media Management Layer (MML) with RMAN

- Oracle maintains the Oracle Backup Solutions Program (BSP). The partners are listed in http://otn.oracle.com/deploy/availability/htdocs/bsp.htm#MMV
- Some of the important one are Legato Systems (NetWorker) and VERITAS (NetBackup).

Obtaining Information about and related to RMAN using Dictionary Views

```
/* archives log files */
-- created, backed up, and cleared in the database
select
 RECID,
 NAME,
 DEST_ID,
 THREAD#,
 SEQUENCE#,
 RESETLOGS_CHANGE#,
 RESETLOGS_TIME,
 RESETLOGS_ID,
 FIRST_CHANGE#,
 FIRST_TIME,
 NEXT_CHANGE#,
 NEXT_TIME,
 round(BLOCKS * BLOCK_SIZE/1024/1024,2) MB ,
 CREATOR,
 ARCHIVED,
 DELETED,
 STATUS,
 COMPLETION_TIME,
 END_OF_REDO,
 BACKUP_COUNT,
 ARCHIVAL_THREAD#,
 IS_RECOVERY_DEST_FILE,
 COMPRESSED,
 FAL,
 BACKED_BY_VSS
from V$ARCHIVED_LOG ;


/* Backup Sets */
select
RECID,
STAMP,
SET_STAMP,
SET_COUNT,
BACKUP_TYPE,
CONTROLFILE_INCLUDED,
INCREMENTAL_LEVEL,
PIECES,
START_TIME,
COMPLETION_TIME,
ELAPSED_SECONDS,
```

```
BLOCK_SIZE,
INPUT_FILE_SCAN_ONLY,
KEEP,
KEEP_UNTIL,
KEEP_OPTIONS,
MULTI_SECTION
from V$BACKUP_SET
order by RECID;


/* Backup Pieces*/
SELECT RECID,
STAMP,
SET_STAMP, -- link to V$BACKUP_SET
SET_COUNT,
PIECE#,
COPY#,
DEVICE_TYPE,
HANDLE,
COMMENTS,
MEDIA,
MEDIA_POOL,
CONCUR,
TAG,
STATUS,
START_TIME,
COMPLETION_TIME,
ELAPSED_SECONDS,
DELETED,
BYTES,
IS_RECOVERY_DEST_FILE,
RMAN_STATUS_RECID,
RMAN_STATUS_STAMP,
COMPRESSED,
BACKED_BY_VSS,
ENCRYPTED,
BACKED_BY_OSB,
FROM V$BACKUP_PIECE
ORDER BY SET_STAMP;


/* Files in Backup Sets */
-- control files and datafiles in backup sets from the control file
select
RECID,
STAMP,
SET_STAMP,
SET_COUNT BACKUPSET_COUNT,
B.FILE#,
d.NAME DATAFILE_NAME,
B.CREATION_CHANGE#,
B.CREATION_TIME,
RESETLOGS_CHANGE#,
RESETLOGS_TIME,
INCREMENTAL_LEVEL,
INCREMENTAL_CHANGE#,
B.CHECKPOINT_CHANGE#,
```

```
B.CHECKPOINT_TIME,
ABSOLUTE_FUZZY_CHANGE#,
MARKED_CORRUPT,
MEDIA_CORRUPT,
LOGICALLY_CORRUPT,
DATAFILE_BLOCKS,
B.BLOCKS,
B.BLOCK_SIZE,
OLDEST_OFFLINE_RANGE,
COMPLETION_TIME,
CONTROLFILE_TYPE,
USED_CHANGE_TRACKING,
BLOCKS_READ,
USED_OPTIMIZATION,
B.FOREIGN_DBID,
B.PLUGGED_READONLY,
B.PLUGIN_CHANGE#,
B.PLUGIN_RESETLOGS_CHANGE#,
B.PLUGIN_RESETLOGS_TIME,
SECTION_SIZE,
UNDO_OPTIMIZED
from V$BACKUP_DATAFILE b, V$DATAFILE d
where B.FILE# (+)= D.FILE#
ORDER BY b.CREATION_TIME DESC;


/* Archive Log files in the Backup Sets */
SELECT
RECID,
STAMP,
SET_STAMP,
SET_COUNT,
THREAD#,
SEQUENCE#,
RESETLOGS_CHANGE#,
RESETLOGS_TIME,
FIRST_CHANGE#,
FIRST_TIME,
NEXT_CHANGE#,
NEXT_TIME,
round(BLOCKS * BLOCK_SIZE/1024/1024,2) MB ,
TERMINAL
FROM V$BACKUP_REDOLOG
ORDER BY RECID DESC;

/* Corruption */
-- blocks have been found corrupt during a backup of a backup set
select * from V$BACKUP_CORRUPTION;

-- database blocks that were corrupted after the last backup
select * from  V$DATABASE_BLOCK_CORRUPTION
```

Starting RMAN

```
export ORACLE_SID=mydb
rman target /
```

```
rman target sys/psw cmdfile D:\..\dblevel0.ora log D:\..\dblevel0.log append
rman target / @myrmanscript.ora
rman target orcl catalog rman/rman@nick

# log parameter doesn’t display output to you. To work around:
rman | tee /u01/app/oracle/rman.log

# using dynamic script: passing parameters to command file
# command file may contain:
BACKUP DATABASE TAG &1
FORMAT '/u02/oracle/bck/&2%U.bck'
# the script running rman may contain:
export format=$2
export mytag=$3
rman @'/u01/app/oracle/scripts/my_backup.cmd' USING  $format $mytag
# you may run the script as:
myscript.sh db01012010 HR01012010
```

Using rlwrap Utility with RMAN in Unix-Based Systems

```
/* Install */
# Option 1: for rpm version:
# can be downloaded from:
# http://rpm.pbone.net
# http://ivan.kartik.sk
rpm -ivh rlwrap*.rpm


# Option 2: for tar version
# download rlwrap-0.30.tar.gz (search the net or from
http://www.ahmedbaraka.com/download/oracle/rlwrap-0.30.tar.gz )
# unzip the file and install
su -
gunzip rlwrap-0.30.tar.gz
tar -xvf rlwrap-0.30.tar
cd rlwrap-0.30
./configure
make
make install

#
echo "alias rman2='rlwrap rman'" >> /home/oracle/.bashrc
```

Configuring the RMAN Environment

```
/* Handling Configuration in General */
# To list current configuration
SHOW ALL
# To set value for a configuration setting
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/?/%U';
CONFIGURE CONTROL FILE AUTOBAKCUP ON
# To reset the setting value
CONFIGURE CONTROL FILE AUTOBAKCUP CLEAR
```

```
/* Backup Retention Policy */
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 14 DAYS;


/* Default Device Type */
CONFIGURE DEFAULT DEVICE TYPE TO sbt;


/* Encryption */
-- Transparent or Dual mode
CONFIGURE ENCRYPTION FOR DATABASE ON


/* Compression */
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COMPRESSED BACKUPSET


/* Defining Parallelism */
CONFIGURE DEVICE TYPE DISK PARALLELISM 4;


/* Configuring Default Channels */
-- number of channels to use depends on DISK PARALLELISM
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT
'/u01/oracle/oradata/backup/%U';
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT
'/u02/oracle/oradata/backup/%U';


/* Backup Optimization */
-- datafile of a version identical to its backup won't be backuped up
CONFIGURE BACKUP OPTIMIZATION ON;


/* Control File Auto-Backup */
CONFIGURE CONTROLFILE AUTOBACKUP ON;
-- Control File Backup Location and Format
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO
'/u01/oracle/oradta/backup/cf_%F';


/* Archivelog Deletion Policy */
-- all destinations are affected including flash recovery area
CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP 2 TIMES TO SBT;
```

RMAN Channel Commands

```
# Manual Channel Allocation
RUN {
ALLOCATE CHANNEL c1 TYPE disk
FORMAT = '/db01/BACKUP/usr0520.bak';
BACKUP DATAFILE '/db01/ORADATA/users01.dbf';}

# Automatic Channel Allocation
CONFIGURE DEFAULT DEVICE TO DISK | SBT ;
```

```
CONFIGUE DEVICE TYPE DISK PARALLELISM n ;

# Automatic Channel Options
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT = '/BACKUP/RMAN/%U' ;
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 2G ;
```

Duration in days of RMAN information in Control File

```
--in days
select value from v$parameter where
upper(name)='CONTROL_FILE_RECORD_KEEP_TIME';

alter system set CONTROL_FILE_RECORD_KEEP_TIME=30 ;
```

Monitoring RMAN Jobs

```
SELECT SID, RECID, STAMP, PARENT_RECID, PARENT_STAMP, SESSION_RECID,
  SESSION_STAMP, ROW_LEVEL, ROW_TYPE, COMMAND_ID, OPERATION, STATUS,
  MBYTES_PROCESSED, START_TIME, END_TIME
from V$RMAN_STATUS
ORDER BY RECID DESC;

# to view message generated by the RMAN operation
select SID, RECID, STAMP, SESSION_RECID, SESSION_STAMP, OUTPUT
from V$RMAN_OUTPUT;
```

Using RMAN BACKUP Command

- BACKUP command options
  FULL
  INCREMENTAL LEVEL n
  INCLUDE CURRENT CONTROLFILE
  FILESPERSET n # maximum number of input files in each backup set
  SKIP OFFLINE | READONLY | INACCESSIBLE
  MAXSETSIZE n K|M|G
  DELETE INPUT # applicable on archived logs, datafile copies or backup sets.
  FORMAT :
    %c copy number
    %p backup piece number
    %s backup set number
    %d database name
    %n database name padded with 8 characters
    %t backup set time stamp
    %u compressed version of %s and %t
    %U (default) equivalent t%u_%p_%c

```
/* Backup Piece Size */
ALLOCATE CHANNEL … MAXPIECESIZE = integer
CONFIGURE CHANNEL … MAXPIECESIZE = integer
```

```
/* Backup examples */
-- make a whole database backup
BACKUP DATABASE FORMAT '/tmp/%U' TAG='weekly_bak'
SQL 'ALTER SYSTEM ARCHIVE LOG CURRENT';

-- Backing Up Tablespaces
BACKUP TABLESPACE system, users, tools;

-- Backing up Datafiles and Datafile Copies
BACKUP DATAFILE 1,2,3,4, DATAFILECOPY '/tmp/system01.dbf'

-- Backing Up Backup Sets (from disk to tape or from disk tdisk)
BACKUP DEVICE TYPE sbt BACKUPSET ALL;
```

Backing Up Control File and SPFile

```
/* Control File Autobackup */
CONFIGURE CONTROLFILE AUTOBACKUP enable ;

/* Control File Backup Format */
SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE disk TO 'controlfile_%F';
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT '..' ;

/* Backing Up the Control File Manually */
BACKUP CURRENT CONTROLFILE TAG='mondaypmbackup' ;

/* Including the Control File in a Backup Set */
BACKUP TABLESPACE users INCLUDE CURRENT CONTROLFILE;

/* Backing Up the Server Parameter File */
-- Automatically backed up when CONFIGURE CONTROLFILE
AUTOBACKUP = ON
-- Explicitly
BACKUP SPFILE
```

Backing Up Archived RedLogs

```
/* Using BACKUP ARCHIVELOG command */
BACKUP ARCHIVELOG ALL ;
BACKUP ARCHIVELOG NOT BACKED UP 2 TIMES;

-- To delete backed up copy of the archived log file
BACKUP ARCHIVELOG ALL DELETE INPUT ;

-- To delete logs from all enabled archiving destinations.
BACKUP ARCHIVELOG ALL DELETE ALL INPUT ;

-- To specify a range of archived redlogs by time
BACKUP ARCHIVELOG UNTIL TIME 'SYSDATE-7';
BACKUP ARCHIVELOG FROM TIME 'SYSDATE-30' UNTIL TIME 'SYSDATE-7';

-- To specify a range of archived redlogs by SCN
BACKUP ARCHIVELOG UNTIL SCN = 320 ;
BACKUP ARCHIVELOG SCN BETWEEN 205 AND 320 ;
```

```
-- To specify a range of archived redlogs by log sequence number
BACKUP ARCHIVELOG UNTIL SEQUENCE = 501
BACKUP ARCHIVELOG FROM SEQUENCE integer

/* Using BACKUP ... PLUS ARCHIVELOG: */
BACKUP DATABASE PLUS ARCHIVELOG;
```

Backup in NOARCHIVELOG Mode

```
1. Shut down cleanly
2. Mount the database.
3. Allocate multiple channels, if not using automatic.
4. Run the BACKUP command.
```

Encrypting RMAN Backups

- Three possible encryption modes for your backups:
  - Transparent mode: It requires Oracle Wallet. It is the default encryption mode.

    `CONFIGURE ENCRYPTION FOR DATABASE ON`
  - Password mode: It requires you to provide a password. It is best suited for backups restored at remote locations.

    `SET ENCRYPTION ON IDENTIFIED BY password ONLY`
  - Dual mode: It can use either Oracle Wallets or passwords.
    After making sure the wallet is open:

    `SET ENCRYPTION ON IDENTIFIED BY password`

    If there is no wallet or the wallet is closed:

    `SET DECRYPTION IDENTIFIED BY password1 {, password2,…, passwordn}`

Using Compression in RMAN Backups

- Only Backupsets can be compressed:

```
RMAN> CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COMPRESSED BACKUPSET
RMAN> BACKUP AS COMPRESSED BACKUPSET DATABASE
```

Using Multiplexed Backup Sets

- RMAN can read from two datafiles simultaneously, and then combine the blocks from these datafiles into a single backup piece.
- Multiplexing can be controlled by the following:
  - The FILESPERSET parameter on the BACKUP command
  - The MAXOPENFILES parameter of the ALLOCATE CHANNEL and CONFIGURE CHANNEL commands (default is 8)

Using Parallelization of Backup Sets

- Parallelization of backup sets is achieved by:
  - Configuring PARALLELISM to greater than 1

- allocating multiple channels
- Specifying many files in the BACKUP command

```
BACKUP
(DATAFILE 1,2,3
 FILESPERSET = 1
 CHANNEL ORA_DISK_1)
(DATAFILECOPY '/tmp/system01.dbf',
 '/tmp/tools01.dbf'
 FILESPERSET = 2
 CHANNEL ORA_DISK_2);

RUN
{
 ALLOCATE CHANNEL c1 DEVICE TYPE sbt PARMS="ENV=
  (BACKUP_SERVER=tape_server1)";
 ALLOCATE CHANNEL c2 DEVICE TYPE sbt PARMS="ENV=
  (BACKUP_SERVER=tape_server2)";
 BACKUP
 (DATAFILE 1,2,3 CHANNEL c1)
 (DATAFILECOPY '/tmp/system01.dbf',
  '/tmp/tools01.dbf' FILESPERSET = 2 CHANNEL c2)
 }
```

Using Duplexed Backup Sets (Backupset Copies)

- CONFIGURE . . . BACKUP COPIES option (maximum 4)
- SET BACKUP COPIES in a RUN block.
- The COPIES option in the BACKUP command.

```
-- BACKUP COPIES
BACKUP DEVICE TYPE DISK COPIES 2 DATAFILE 1 FORMAT '/disk1/df1_%U',
'/disk2/df1_%U';
BACKUP COPIES 2 DATAFILE 1, DATAFILE 2 FORMAT '/BACKUP1/%U','/BACKUP2/%U';
BACKUP COPIES 3 INCREMENTAL LEVEL = 0 DATABASE;

-- CONFIGURE … BACKUP COPIES
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/save1/%U', '/save2/%U';
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE sbt T2;
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE sbt T2;
```

Making Image Copies

- BACKUP AS COPY..

```
BACKUP AS COPY DATABASE;
BACKUP AS COPY TABLESPACE SYSAUX;
BACKUP AS COPY DATAFILE 2;

BACKUP AS COPY COPY OF DATABASE;

BACKUP AS BACKUPSET COPY OF TABLESPACE SYSAUX;

BACKUP AS COPY COPY OF DATAFILE 2;

-- to check
```

```
list backupset of tablespace tbs1;
```

- In 9i, (**DEPRICATED IN 10G)**

```
COPY DATAFILE { 'filename'| integer }
  | DATAFILECOPY {'filename' |TAG='tag_name'}
  | ARCHIVELOG 'filename'
  | CURRENT CONTROLFILE
  | CONTROLFILECOPY {'filename'|TAG='tag_name'}
    TO AUXNAME | 'filename'
```

```
COPY DATAFILE '/ORADATA/users_01_db01.dbf' to '/BACKUP/users01.dbf' tag=DF3 ;
```

Validating Backup

- When you run BACKUP VALIDATE, RMAN checks datafiles for physical and logical block corruption but it does not actually produce any backup sets or image copies.

```
# only database level
BACKUP VALIDATE DATABASE ARCHIVELOG ALL;

# db, tablespace, datafile, block levels
VALIDATE BACKUPSET 1;
```

Incremental Backup

- Level 0 or Level 1.
- Differential: since last 1 or 0, Cumulative: since last 0 (faster recovery).

```
BACKUP INCREMENTAL LEVEL N [CUMULATIVE]
```

backup of: DATAFILE, DATAFILECOPY, TABLESPACE, or DATABASE.

```
BACKUP INCREMENTAL LEVEL 0 ...
```

Tags for Backups and Image Copies

```
BACKUP .. TAG='tag_name'
```

Creating Archival Backups

- The BACKUP … KEEP command can be used to create a backup that is both all-inclusive (every file needed including archived redo logs) and exempt from the backup retention policy.
- In Oracle 11g, some modifications made on the RMAN BACKUP … KEEP command. In the new version of the command, the KEEP, NOKEEP, FOREVER, and UNTIL TIME options are retained. However, the LOGS and NOLOGS options are not there any longer. Instead, you have a new option, RESTORE POINT. The RESTORE POINT option lets RMAN automatically create a normal restore point.

Note: You can't use the KEEP clause for backup files in the flash recovery area. Also, you cannot use the CHANGE … KEEP command for backup files stored in the flash recovery area.

```
RUN
```

```
{
 ALLOCATE CHANNEL c1 DEVICE TYPE sbt PARMS
'ENV=(OB_MEDIA_FAMILY=archival_backup)';
 -- with forever option (recovery catalog is required)
 BACKUP DATABASE TAG BAKQ108 KEEP FOREVER RESTORE POINT FY08Q1;
 -- backup will be kept for 365 days (long-term)
 BACKUP DATABASE TAG BAKQ108 KEEP UNTIL TIME 'SYSDATE+365' RESTORE POINT
FY08Q1;
 -- After one day, the backup becomes obsolete,
 --  regardless the configured retention policy
 BACKUP DATABASE FORMAT '/u01/oraclebck/%U.bck'
   TAG TESTDB KEEP UNTIL 'SYSDATE+1' RESTORE POINT TESTDB08;
}
```

- If you want to change the status of a regular backup to an archival backup, use the CHANGE command as follows:

```
CHANGE BACKUP TAG 'weekly_bkp' KEEP FOREVER;
-- make it follow back the retention policy
CHANGE BACKUP TAG 'weekly_bkp' NOKEEP;
```

Monitoring RMAN Backups

To correlate a process with a channel during a backup:

1. In each session, set the COMMAND ID to a different value

```
RUN
{
ALLOCATE CHANNEL c1 TYPE sbt;
SET COMMAND ID TO 'sess1';
BACKUP DATABASE;
}
```

2. Query the joined V$SESSION and V$PROCESS views

```
SELECT SID, SPID, CLIENT_INFO
FROM V$PROCESS p, V$SESSION s
WHERE p.ADDR = s.PADDR
AND s.CLIENT_INFO LIKE '%id=sess%';
```

The CLIENT_INFO column displays in the following format:

```
id=command_id,rman channel=channel_id
```

3. Query the V$SESSION_LONGOPS view to get the status of the backup or copy

```
SELECT s.SID, S.SERIAL#, CLIENT_INFO, round(L.ELAPSED_SECONDS/60,2)
ELAPSED_TIME_MIN ,L.SOFAR, L.TIME_REMAINING/60 TIME_REMAINING_MIN
FROM V$PROCESS p, V$SESSION s, V$SESSION_LONGOPS l
WHERE p.ADDR = s.PADDR and L.SID = s.SID and L.SERIAL# = s.SERIAL#
AND s.CLIENT_INFO LIKE '%rman%'
AND NVL(L.TIME_REMAINING,0)<>0;
```

RMAN Complete Recovery

Validating Backup Files

- You can use VALIDATE Validate backup sets before you use them from a recovery.
- You can run RESTORE … VALIDATE to test whether RMAN can restore a specific file or set of files from a backup. RMAN chooses which backups to use.

```
LIST COPY;
LIST BACKUP;
LIST BACKUP RECOVERABLE;

VALIDATE BACKUPSET 1;

-- no restore is actually done
--  lack of errors: validation found no problem
RESTORE DATABASE VALIDATE;
RESTORE ARCHIVELOG ALL VALIDATE;
```

Previewing Backup Files Required by a Restore

- The RESTORE . . . PREVIEW command provides a detailed report of all backups that are necessary for that RESTORE command to succeed.

```
RESTORE DATABASE PREVIEW;
RESTORE DATABASE PREVIEW SUMMARY;
RESTORE TABLESPACE users PREVIEW;
RESTORE DATAFILE 3 PREVIEW;
```

Identifying Datafiles Requiring Recovery

```
SELECT r.FILE# AS df#, d.NAME AS df_name, t.NAME AS tbsp_name,
 d.STATUS, r.ERROR, r.CHANGE#, r.TIME
FROM V$RECOVER_FILE r, V$DATAFILE d, V$TABLESPACE t
WHERE t.TS# = d.TS# AND d.FILE# = r.FILE#;
```

Performing Complete Recovery

```
-- Recover a Database in ARCHIVELOG Mode
STARTUP MOUNT;
RESTORE DATABASE;
RECOVER DATABASE;


-- Restore Datafiles to a New Location
run{
 set newname for datafile 1 to '/newdir/system01.dbf';
 restore database;
 switch datafile all; # record in control file
 recover database;}
-- alternatively:
run {
 sql "alter tablespace tbs1 offline immediate" ;
 set newname for datafile 6 to 'C:\APP\ADMINISTRATOR\ORADATA\ORA11G\tbs1.dbf';
 restore tablespace tbs1;
 switch datafile 6;
 recover tablespace tbs1; }
sql "alter tablespace tbs1 online" ;


-- Recover a Tablespace
run{
 sql "alter tablespace users offline immediate";
 restore tablespace users;
 recover tablespace users;
 sql "alter tablespace users online"; }
```

```
-- Relocate a Tablespace
RUN{
 SQL "alter tablespace users offline immediate";
 SET NEWNAME FOR DATAFILE '/ORADATA/u03/users01.dbf'
  TO '/ORADATA/u04/users01.dbf';
 RESTORE TABLESPACE users;
 SWITCH datafile 3; # Update the control file and recovery catalog
 RECOVER TABLESPACE users; #Recover the tablespace
 SQL "alter tablespace tbs1 online";}
```

Restoring whole Database from RMAN Backups On a Different Node

- The directory structures can be the same or different
- The procedure applies for database in ARCHIVELOG mode or NOARCHIVELOG mode.
- Reference: Document ID 419137.1

```
The following example assume restoring a database from node1 to node2.

/*  Move the following files to the NODE 2 */
+ The database backup pieces to location '/node2/database/backup'
+ Controlfile backup piece to the location '/node2/database/backup'
+ The parameter file i.e init.ora file to the default location:
Unix: $ORACLE_HOME/dbs
Windows: $ORACLE_HOME/database


/* Edit the PFILE on NODE 2 to change the environment
    specific parameters and create thier directories */
user_dump_dest=
background_dump_dest=
control_files=              <--specify the new location of controlfiles
db_recovery_file_dest=


/* set the Oracle environment variables and
   start the database in nomount mode  */
Unix:
export ORACLE_HOME=/u01/oracle/product/ora10g
export ORACLE_SID=ora10g
export PATH=$ORACLE_HOME/bin:$PATH

Windows:
set ORACLE_HOME=D:\oracle\product\10.1.0\db_1
set ORACLE_SID=oradb
set PATH=D:\oracle\product\10.1.0\db_1\bin;%PATH%


/* Create a password file: */
Unix:
$orapwd file=$ORACLE_HOME/dbs/orapw$ORACLE_SID.ora password=*** entries=5
Windows:
orapwd file=%ORACLE_HOME%\database\PWD%ORACLE_SID%.ora password=*** entries=5


/* In Windows: create the SID service: */
```

```
oradim -NEW -SID oradb -STARTMODE auto


/* invoke Rman on the NODE 2 */
-- in Windows, it's already started so shutdown first
rman target /
Recovery Manager: Release 10.2.0.1.0 - Production on Tue Feb 13 00:36:55 2007
Copyright (c) 1982, 2005, Oracle.  All rights reserved.
connected to target database (not started)
RMAN> startup nomount pfile=X:\..\PFILE.ORA
Oracle instance started
Total System Global Area     205520896 bytes
Fixed Size                     1218508 bytes
Variable Size                 75499572 bytes
Database Buffers             121634816 bytes
Redo Buffers                   7168000 bytes


/* Restore the controlfile from the backup piece. */
-- controlfiles will be resotred to locations indicated in pfile
RMAN> restore controlfile from
'D:\temp\backup\ORADB\AUTOBACKUP\2011_07_06\O1_MF_S_755814486_718GDZ6L_.BKP';
Starting restore at 06/07/11
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=380 devtype=DISK

channel ORA_DISK_1: restoring controlfile
channel ORA_DISK_1: restore complete
output filename=D:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\CONTROL01.CTL
output filename=D:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\CONTROL02.CTL
output filename=D:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\CONTROL03.CTL
Finished restore at 06/07/11


/* Mount the database */
RMAN> alter database mount;


/* Now catalog the backup pieces that were shipped from NODE 1 */
RMAN> catalog backuppiece
'/node2/database/backup/o1_mf_annnn_TAG20070213T002925_2x21m6ty_.bkp';
RMAN> catalog backuppiece
'/node2/database/backup/o1_mf_annnn_TAG20070213T002825_2x21kbds_.bkp';
RMAN> catalog backuppiece
'/node2/database/backup/o1_mf_nnndf_TAG20070213T002827_2x21kd12_.bkp';

RMAN>catalog backuppiece
'D:\temp\backup\ORADB\BACKUPSET\2011_07_06\O1_MF_NNNDF_TAG20110706T203113_718G
CL0P_.BKP';


/* In case of ARCHIVELOGMODE: Get to know the last sequence available in the
  archivelog backup using the following command.*/
-- This will help us in recovering the database till that archivelog.
RMAN > list backup of archivelog all;
Let us assume the last sequence of last archivelog in the backup is 50.
```

```
/* Rename the Redologfiles,so that they can be created in new locations
    the database is opened in resetlogs */
SQL>conn sys as sysdba
-- get registered files from SELECT * FROM V$LOGFILE
SQL> alter database rename file '/node1/database/prod/redo01.log' to
'/node2/database/prod/redo01.log';
....
....

SQL>alter database rename file
alter database rename file 'E:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\REDO01.LOG' to
 'D:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\REDO01.LOG';
....
....


/* Now restore the datafiles to new locations and recover. */
Since we are recovering the database here till the archivelog sequence 50 the
sequence number in the SET UNTIL SEQUENCE clause should be  50
Note: If we are restoring the Rman backups from tapes, then we should ensure
the same media manager variables that were used during backups are maintained
during restore too. The following document articles gives the information of
various media manager Environment Variables:
NOTE.312737.1 RMAN and Specific Media Managers Environment Variables.

Note: in NOARCHIVEMOE, omit recover database below
RMAN> run {
 set until sequence 51;
 set newname for datafile 1 to '/node2/database/prod/sys01.dbf';
 set newname for datafile 2 to '/node2/database/prod/undotbs01.dbf';
 set newname for datafile 3 to '/node2/database/prod/sysaux01.dbf';
 set newname for datafile 4 to '/node2/database/prod/users01.dbf';
 set newname for datafile 5 to '/node2/database/prod/1.dbf';
 set newname for datafile 6 to '/node2/database/prod/sysaux02.dbf';
 set newname for datafile 7 to '/node2/database/prod/undotbs02.dbf';
 restore database;
 switch datafile all;
 recover database;
 alter database open resetlogs;
 }


Note: in NOARCHIVEMOE, omit recover database below
RMAN> run {
   set newname for datafile 'E:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\SYSTEM01.DBF'
to 'D:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\SYSTEM01.DBF';
   set newname for datafile
'E:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\UNDOTBS01.DBF'
to 'D:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\UNDOTBS01.DBF';
   set newname for datafile 'E:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\SYSAUX01.DBF'
to 'D:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\SYSAUX01.DBF';
   set newname for datafile 'E:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\USERS01.DBF'
to ‘D:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\USERS01.DBF';
   set newname for datafile 'E:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\SATBS01.DBF'
to 'D:\ORACLE\PRODUCT\10.1.0\DB_1\ORADB\SATBS01.DBF';
 restore database;
 switch datafile all;
 alter database open resetlogs;
```

```
}


/* Addd tempfile to temporary tablespaces */
-- list of temporary tablespaces that require datafiles are in alert file
ALTER TABLESPACE temp ADD TEMPFILE ' ..' SIZE 500M AUTOEXTEND ON NEXT 100M
MAXSIZE 8G;


/* Some recommended post-restore */
-- user created jobs will automatically run if their scheduled date is due
-- you may consider stopping them
as job owner:
-- to list of the jobs:
select JOB, BROKEN, SCHEMA_USER, NEXT_DATE, WHAT
from dba_jobs;

begin
 dbms_job.broken( 387 ,true);
 commit;
end;
/
```

Restoring whole Database from RMAN Backups from A 32 bit to 64 bit

- Reference: Document ID 467676.1

```
-- follow all the steps in the pervious section titled "Restoring whole
Database from RMAN Backups On a Different Node" till you reach to the step
"Now restore the datafiles to new locations and recover."

-- implement the steps in that step but change opening statement as follows:
ALTER DATABASE OPEN RESETLOGS MIGRATE;

-- all PL/SQL objects should be re-compiled
-- after opening the database with resetlogs:
-- invalidate and all pl/sql modules
SQL> @ ?/rdbms/admin/utlirp.sql

SQL> shutdown immediate;
SQL> startup;
-- recompiles invalid objects in the database
SQL> @ ?/rdbms/admin/utlrp.sql

SQL> shutdown immediate;
SQL> startup
```

RMAN Incomplete Recovery

1. Mount the database.

2. The following steps should be followed:

```
RUN { # multiple channels for parallelization
    ALLOCATE CHANNEL c1 TYPE DISK;
    ALLOCATE CHANNEL c2 TYPE DISK; # recover until time, SCN or sequence
    SET UNTIL TIME = '2001-12-09:11:44:00'; SET UNTIL TIME "to_date('09-05-2004
    00:00:20', 'dd-mm-yyyy hh24:mi:ss')" ; SET UNTIL SEQUENCE 120 THREAD 1;#120 not
    included
```

```
    # ALL datafiles must be restored
    RESTORE DATABASE;
    RECOVER DATABASE;
ALTER DATABASE OPEN RESETLOGS; }
```

3. If using a recovery catalog, register the new incarnation of the database using the command: `RESET DATABASE`

4. Perform a whole database backup.

Note: Insure that `NLS_LANG` and `NLS_DATE_FORMAT` environment variables are set appropriately.

Note: check the `alert.log` for any errors during recovery.

Note: If you need to restore archived redo log files to a new location use the `SE SET ARCHIVELOG DESTINATION TO <location>` command.

Simplified Recovery Through Resetlogs

- Target: to recovery through current and ancestor database incarnations.
- You must reset the current incarnation of the database back to the incarnation to which your target SCN belongs.
- You must use the control file from the older incarnation that contains the target SCN.

```
Find out the incarnation key for the incarnation that was current at the time
you want to recover your database to.
RMAN>LIST INCARNATION

Start the database in the following way:
RMAN> STARTUP FORCE NOMOUNT;

Reset the current incarnation:
RMAN> RESET DATABASE TO INCARNATION 2;

Restore the old control file from a backup:
RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;
RMAN> ALTER DATABASE MOUNT;

Restore and recover the database:
RMAN> RESTORE DATABASE;
RMAN> RECOVER DATABASE UNTIL SCN 1000;

Open the database:
RMAN> ALTER DATABASE OPEN RESETLOGS;
```

Recovering from Lost Control File using RMAN

```
If database isn't already shutdown:
RMAN>shutdown abort

RMAN>startup nomount

RMAN>RESTORE CONTROLFILE FROM AUTOBACKUP;

ALTER DATABASE MOUNT;

RECOVER DATABASE;
```

```
ALTER DATABASE OPEN RESETLOGS;
```

Block Media Recovery (BMR)

- RMAN's BLOCKRECOVER command recovers blocks marked as corrupt in the V$BACKUP_CORRUPTION and V$COPY_CORRUPTION views.
- BMR helps on the following goals:
  - Faster recovery time.
  - Increased database availability.
- BMR can repair only physically corrupted blocks but not the logically corrupted blocks.

```
-- typical scenario:
ORA_11578: ORACLE data block corrupted (file# 9, block# 21)
ORA=01110: data file 9: /u01/app/oracle/oradata/remorse/users_01.dbf'

-- the command will search flashback logs then backup files
-- specific block
RECOVER DATAFILE 9 BLOCK 21;

-- all blocks in V$DATABASE_BLOCK_CORRUPTION
RECOVER CORRUPTION LIST;
-- to query which blocks recovered:
SELECT * FROM V$DATABASE_BLOCK_CORRUPTION;
```

Trial Recovery

- Target: to estimate size of corruption in a recovery process.
- It lets you know whether there is corruption and, if there is, the extent of the corruption.

```
-- You can use the TEST option for any RECOVER command
RECOVER DATABASE TEST
RECOVER DATABASE USING BACKUP CONTROLFILE UNTIL CANCEL TEST
RECOVER TABLESPACE users TEST
RECOVER DATABASE UNTIL CANCEL TEST

-- You can limit the number of data blocks trial recovery can corrupt in
memory:
RECOVER DATABASE TEST ALLOW 10 CORRUPTION;

SQL> RECOVER DATABASE UNTIL CANCEL TEST;
ORA-10574: Test recovery did not corrupt any data block
ORA-10573: Test recovery tested redo from change 9948095 to 9948095
ORA-10570: Test recovery complete
```

Handling Specific Errors During Recovery

```
ORA-01194 Error:
----------------
-- this error may raise during startup a cloned database
-- resolution: provide the online redo log file to recover

SQL> startup
```

```
ORACLE instance started.
..
Database mounted.
ORA-01589: must use RESETLOGS or NORESETLOGS option for database open
SQL> alter database open noresetlogs;
alter database open noresetlogs
*
ERROR at line 1:
ORA-01588: must use RESETLOGS option for database open
SQL> alter database open resetlogs;
alter database open resetlogs
*
ERROR at line 1:
ORA-01194: file 1 needs more recovery to be consistent
ORA-01110: data file 1: 'C:\ORACLE\ORADATA\MANAGER\SYSTEM01.DBF'

SQL> RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE;
ORA-00279: change 405719 generated at 06/30/2008 15:51:04 needed for thread 1
ORA-00289: suggestion : C:\ORACLE\RDBMS\ARC00019.001
ORA-00280: change 405719 for thread 1 is in sequence #19
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
C:\ORACLE\ORADATA\MANAGER\REDO03.LOG
Log applied.
Media recovery complete.
SQL> alter database open resetlogs;
Database altered.



ORA-01152 Error:
----------------
-- resolution: provide the online redo log file to recover

ORA-00289: suggestion :
/u01/app/oracle/admin/finance/arch/finance/_0000012976.arc
ORA-00280: change 962725326 for thread 1 is in sequence #12976
ORA-00278:
logfile'/u01/app/oracle/admin/finance/arch/finance/_0000012975.arc'
no longer needed for this recovery
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-01547: warning: RECOVER succeeded but OPEN RESETLOGS would get error below
ORA-01152: file 1 was not restored from a sufficiently old backup
ORA-01110: data file 1: '/pase16/oradata/finance/system_01.dbf'ORA-01112:
media recovery not started
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
/pase04/oradata/finance/redo01a.rdo
ORA-00279: change 962746677 generated at 07/30/2008 04:33:52 needed for thread
1
ORA-00289: suggestion :
/u01/app/oracle/admin/finance/arch/finance/_0000012978.arc
ORA-00280: change 962746677 for thread 1 is in sequence #12978
ORA-00278: log file '/pase04/oradata/finance/redo01a.rdo'
no longer needed for this recovery
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
/pase04/oradata/finance/redo02a.rdo
Log applied.
Media recovery complete.
```

```
ORA-00376 Error:
----------------
reason: might be datafile or tablespace being offline
resolution: bringing the tablespace or datafile online

ORA-00376: file 10 cannot be read at this time
ORA-01110: data file 10: '/u01/app/oracle/remorse/data_01.dbf'
```

Configuring Instance Crash Recovery Time (MTTR)

```
SELECT VALUE FROM V$PARAMETER WHERE UPPER(NAME)='FAST_START_MTTR_TARGET';

-- in seconds
ALTER DATABASE SET FAST_START_MTTR_TARGET=60;

SELECT recovery_estimated_ios, estimated_mttr, target_mttr
FROM v$instance_recovery;
```

Working with the Data Recovery Advisor in RMAN

```
list failure;
list failure critical;
LIST FAILURE HIGH;
LIST FAILURE LOW;
list failure open;
list failure closed;
list failure exclude failure 12345;

advise failure;
repair failure preview;


repair failure;
The V$IR_REPAIR view shows the results of the REPAIR FAILURE command:
select REPAIR_ID,ADVISE_ID,SUMMARY,RANK from V$IR_REPAIR;
```

RMAN Maintenance

Cross Checking Backups and Copies

- Use CROSSCHECK command to ensure that data about backup sets and image copies in the recovery catalog or control file is synchronized with corresponding files on disk or in the media management catalog.

```
CROSSCHECK BACKUPSET OF DATABASE;
CROSSCHECK BACKUP OF TABLESPACE users DEVICE TYPE sbt COMPLETED BEFORE
'SYSDATE-31';
Page 11 Oracle Database Administration Fundamentals II (Note Sheets)
CROSSCHECK BACKUP OF ARCHIVELOG ALL SPFILE;
```

```
CROSSCHECK BACKUPSET 1338, 1339, 1340;
CROSSCHECK BACKUPPIECE TAG = 'nightly_backup';
CROSSCHECK CONTROLFILECOPY '/tmp/control01.ctl';
CROSSCHECK DATAFILECOPY 113, 114, 115;
Note: If the backup or copy is no longer available, then RMAN marks it as
EXPIRED. You can determine which files are marked EXPIRED by issuing a LIST
EXPIRED command.
```

Deleting Backups and Copies

```
/* Deleting Specified Backups and Copies  */
DELETE BACKUPPIECE 101;
DELETE CONTROLFILECOPY '/tmp/control01.ctl';
DELETE NOPROMPT ARCHIVELOG UNTIL SEQUENCE = 300;
DELETE BACKUP OF TABLESPACE users DEVICE TYPE sbt;
DELETE COPY OF CONTROLFILE LIKE '/tmp/%';
DELETE NOPROMPT BACKUP OF SPFILE COMPLETED BEFORE 'SYSDATE-7';
DELETE NOPROMPT ARCHIVELOG ALL BACKED UP 3 TIMES TO sbt;

/* Deleting Expired Backups and Copies */
-- files are marked as EXPIRED when not found by CROSSCHECK
DELETE EXPIRED BACKUP; -- REPORT OBSOLETE
DELETE EXPIRED COPY;
DELETE NOPROMPT EXPIRED BACKUP OF TABLESPACE users
DEVICE TYPE sbt COMPLETED BEFORE 'SYSDATE-31';

/* Deleting Backups and Copies Rendered Obsolete by the Retention Policy */
DELETE OBSOLETE;

/* Deleting Backups and Copies Defined as Obsolete */
DELETE OBSOLETE REDUNDANCY = 3;
DELETE OBSOLETE RECOVERY WINDOW OF 7 DAYS;

/* Forcing the Deletion of Backups and Copies */
DELETE FORCE NOPROMPT BACKUPSET TAG 'weekly_bkup';
```

Changing the Availability of RMAN Backups and Copies

- If a file is marked UNAVAILABLE, RMAN will not use the file when a RESTORE or RECOVER command is issued.
- Use the command CHANGE … UNAVAILABLE

```
CHANGE DATAFILECOPY '/DB01/BACKUP/users01.dbf' UNAVAILABLE;
CHANGE BACKUP OF CONTROLFILE UNAVAILABLE;
CHANGE BACKUP OF CONTROLFILE AVAILABLE;
CHANGE COPY OF ARCHIVELOG SEQUENCE BETWEEN 230 AND 240 UNAVAILABLE;
```

Exempting a Backup or Copy from the Retention Policy

- Note: Specify KEEP … LOGS to save archived logs for a possible incomplete recovery and KEEP … NOLOGS not to save archived logs for a possible incomplete recovery.
- Note: The KEEP FOREVER clause requires the use of a recovery catalog.
- Note: Use CHANGE … NOKEEP to make the file conform to the retention policy.
- Use the command CHANGE … KEEP FOREVER|UNTIL

```
CHANGE BACKUPSET 123 KEEP FOREVER NOLOGS;
CHANGE DATAFILECOPY '/DB01/BACKUP/users01.dbf' KEEP UNTIL 'SYSDATE+60';
```

The CATALOG Command

- Use CATALOG command to make RMAN aware of the existence of objects that are not recorded in the repository.
- You need to make RMAN aware of the existence of archived redo log files that are not recorded in the repository, if you manually have restored your control file from a backup.

```
CATALOG DATAFILECOPY '/DB01/BACKUP/users01.dbf';
CATALOG CONTROLFILECOPY ‘/DB01/BACKUP/db.ctl’;
CATALOG ARCHIVELOG '/ORADATA/ARCHIVE1/arch_12.arc',
'/ORADATA/ARCHIVE1/arch_13.arc';

# start searching for all uncataloged files in the directory
CATALOG START WITH '/disk1/backups/';

# Catalog all files and the contents of directories which
# begin with the pattern "/backup/MYSID/arch".
CATALOG START WITH '/backup/MYSID/arch';

# Catalog all files in the current recovery area.
CATALOG RECOVERY AREA NOPROMPT;

# Catalog all files in the current recovery area.
# This is an exact synonym of the previous command.
CATALOG DB_RECOVERY_FILE_DEST NOPROMPT;
```

The CHANGE … UNCATALOG Command

- Run the CHANGE … UNCATALOG command to perform the following actions on RMAN repository records:
  - Delete a specific backup or copy record from the recovery catalog
  - Update a backup or copy record in the target control file repository to status DELETED
- RMAN does not touch the specified physical files: it only alters the repository records for these files. You can use this command when you have deleted a backup or copy through a means other than RMAN.
- The CHANGE … UNCATALOG is not supported for backupsets. Use Run the CHANGE … UNAVAILABLE or CHANGE … DELETE instead.

```
CHANGE ARCHIVELOG … UNCATALOG;
CHANGE DATAFILECOPY '/DB01/BACKUP/users01.dbf' UNCATALOG;
```

RMAN Catalog

Creating a Recovery Catalog

1. Create tablespace

2. Create catalog owner

```
CREATE USER RMAN IDENTIFIED BY rman
 TEMPORARY TABLESPACE temp
 DEFAULT TABLESPACE rman_tbs
 QUOTA UNLIMITED ON rman_tbs;
```

3. Grant privileges

```
GRANT connect, resource, recovery_catalog_owner to rman;
```

4. Create catalog

```
rman catalog rman_db1/rman_db1@catdb

RMAN> create catalog tablespace rman_ts;
```

5. Connect to target database as SYSDBA

```
RMAN TARGET SYS/SYS@TEST2 CATALOG RMAN/RMAN@TEST1
```

6. Register target database (**make sure ORACLE_SID is properly set**):

```
REGISTER DATABASE;
```

RMAN Catalog Reporting

Use `REPORT` and `LIST` commands.

- Alternatively you can use SQL commands to query the RMAN data dictionary views. Examples of those view are the following:

```
RC_DATABASE
RC_DATAFILE
RC_STORED_SCRIPT
RC_STORED_SCRIPT_LINE
RC_TABLESPACE
```

```
/* report command */
REPORT SCHEMA
REPORT OBSOLETE;
REPORT NEED BACKUP;
# datafile containing undergone a nologging data (must be backed up)
REPORT UNRECOVERABLE

/* list command */
LIST BACKUP;
LIST BACKUP RECOVERABLE;
LIST COPY;
LIST ARCHIVELOG ALL;
LIST SCRIPT NAMES;
LIST GLOBAL SCRIPT NAMES;
LIST INCARNATION;
```

Upgrading a Recovery Catalog

- If your RMAN client is from the Oracle 11.1 release, but the recovery catalog schema is from an older version, you must upgrade the recovery catalog.

```
-- to know current version of ur catalog
SELECT * FROM rcver;

-- upgrade steps
1) If the recovery catalog owner that you created is from a release before
10.1, execute the following
GRANT command (assuming that rman is the catalog owner):
SQL> GRANT CREATE TYPE TO rman;
2) Start RMAN and connect to the recovery catalog database.
RMAN> connect catalog rman/rman;
3) Execute the UPGRADE CATALOG command.
RMAN> UPGRADE CATALOG;
4) Confirm the command by rerunning it.
RMAN> UPGRADE CATALOG;
You can now use the recovery catalog with the RMAN client from the Oracle
Database 11g release.
```

Importing Recovery Catalogs

- To merge two recovery catalogs, one from the 10.2 release and the other from 11g, into a single 11g release catalog schema.

```
1. Connect to the destination recovery catalog.
$ rman
RMAN> connect catalog rman/rman@rman11
2. Issue the IMPORT CATALOG command
RMAN> import catalog rman1/rman1@rman10;
To specify which database to register:
RMAN> import catalog rman10/rman10@tenner dbid = 123456, 1234557;
RMAN> import catalog rman10/rman10@tenner db_name = testdb, mydb;
-- by default, the databases are unregistered from the source catalog:
RMAN> import catalog rman10/rman10@tenner NO UNREGISTER
```

Moving a Recovery Catalog

```
-- 1) create empty recovery catalog
-- 2) connect and import:
RMAN> connect catalog rman/rman@target_db
RMAN> import catalog rman10/rman10@source_db;
```

Dropping a Recovery Catalog

```
connect catalog rman/rman@target_db
DROP CATALOG;
```

Virtual Private Catalogs

In Oracle Database 11g, you can restrict access to the recovery catalog by granting access to only a subset of the metadata in the recovery catalog. The subset that a user has read/write access to is termed as *virtual private catalog*, or just *virtual catalog*. The central or source recovery catalog is now called the *base recovery catalog*.

Following are the steps to create a new private catalog for the database user SCOTT:

```
# grant the role RECOVERY_CATALOG_OWNER to the user
SQL>GRANT RECOVERY_CATALOG_OWNER TO scott;
# in RMAN session, connect as the base catalog owner
RMAN>CONNECT CATALOG rman/rman@mydb
RMAN>GRANT CATALOG FOR DATABASE db1, db2 TO SCOTT;
# connect as the granted user (virtual catalog onwer) and create the virtual catalog
RMAN>CONNECT CATALOG scott/lion@mydb
RMAN>CREATE VIRTUAL CATALOG;
# make sure only granted dbs are seen
RMAN>LIST INCARNATION;
```

If the catalog is to be used for releases pre-Oracle 11g clients, in the SQL*Plus log on as the virtual private catalog owner and run the following procedure, where "rman" is the name of the base catalog owner:

```
SQL> CONN scott/lion@mydb
SQL> EXEC rman.DBMS_RCVCAT.CREATE_VIRTUAL_CATALOG;
```

Note A virtual private catalog owner can create a local stored script, but has only read-only access to global scripts.

The CATALOG FOR DATABASE privileges include the privilege to register and unregister those databases for which the catalog for database privilege was granted.

The set of views and synonyms that makes up the virtual private catalog is stored in the schema of the virtual catalog owner.

Managing Virtual Private Catalogs

The base recovery catalog owner can optionally grant a virtual catalog owner the right to register new target databases in the recovery catalog by specifying the REGISTER database clause with the GRANT command. Following is an example:

```
RMAN> grant register database to scott;
```

The virtual catalog owner must have the SYSDBA and SYSOPER privileges on the traget database, to perform most of the RMAN operations on it.

Following are examples of removing the privileges from a virtual catalog owner:

```
# To remove recovery catalog access to a database from a user:
RMAN>CONNECT CATALOG RMAN/RMAN@MYDB;
RMAN>REVOKE CATALOG FOR DATABASE db1 FROM scott;

# To revoke the ability to register new databases from a virtual private catalog owner:
RMAN>REVOKE REGISTER DATABASE FROM scott;

# To revoke both the catalog and register privileges from a user:
RMAN>REVOKE ALL PRIVILEGES FROM scott;
```

Dropping a Virtual Private Catalog

Virtual private catalog owners can drop the private recovery catalog they own by issuing the DROP CATALOG command. Following is an example:

```
# Log in as the virtual catalog owner:
RMAN>CONNECT CATALOG scott/<password>@mydb;

# Issue the drop catalog command
RMAN>DROP CATALOG;
```

Caution When the DROP CATALOG command is issued by the virtual catalog owner, all the metadata pertaining to it is deleted from the base recovery catalog.

Using RMAN Scripts

```
# create script
CREATE SCRIPT nightly_backup {
 ALLOCATE CHANNEL c1 TYPE DISK;
 BACKUP DATABASE FORMAT '/u01/app/oracle/%u';
 SQL 'ALTER DATABASE BACKUP CONTROLFILE TO TRACE';
 }

# run it
RUN {EXECUTE SCRIPT nightly_backup;}

# create global script: accessible by any registered db
CREATE GLOBAL SCRIPT global_full_backup
{..

# creating and using dynamic scripts:
CREATE SCRIPT myscript { ... BACKUP TAG &1 ..
RUN
{ EXECUTE SCRIPT myscript
  USING arch_bkp
   FY09Q4;}

# print a script on the screen ( or query RC_STORED_SCRIPT_LINE )
```

```
PRINT GLOBAL SCRIPT global_full_backup;

# listing script names
LIST SCRIPT NAMES;

# update a script
REPLACE SCRIPT full_backup { ..

# converting stored script to text file
PRINT script nightly_backup to file 'test.txt';

# delete a script
DELETE SCRIPT 'my-script';
```

Duplicating (Cloning) a Database

- A duplicate database is a copy of your target database with a new, unique database identifier (DBID).

Database Duplication (Cloning) Methods

- Using RMAN DUPLICATE command
- Using the OEM Database Control
- Manually with SQL*Plus

Database Duplication Techniques

Following are the techniques for database duplication:

- Duplicating an Active Database: From an active database, connected to the target and auxiliary instances (no backup is required).
- Duplicating a Database with a Target Connection: From backup, connected to the target and auxiliary instances.
- Duplicating a Database with Recovery Catalog Without Target Connection: From backup, connected to the auxiliary instance, not connected to the target, but with recovery catalog connection.
- Duplicating a Database without Recovery Catalog or Target Connection: From backup, connected to the auxiliary instance, not connected to the target and the recovery catalog.

Database Duplication Prerequisites

- Both the target and destination databases must be on an identical operating system platform.
- The password file must exist for the source database and both the target and destination databases must have the same SYS password.
- The target database must be open or in mount state.
- If the target database is open, it must be in ARCHIVELOG mode.

Duplicating an Active Database using RMAN

```
-Create a password file in the destination server:
orapwd FILE=D:\oracle\product\11.1.0\db_1\database\PWDora11g2.ora
PASSWORD=ora11g ENTRIES=10 ignorecase=n

-Establish Oracle Net connectivity to the auxiliary instance in both the
source and destination servers:
ORA11G2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = moep10)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
```

```
      (SERVICE_NAME = ora11g2)
    )
  )


-Add the auxiliary database service to the listener configuration file in the
source server:
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = ora11g)
      (ORACLE_HOME = C:\app\Administrator\product\11.1.0\db_1)
      (SID_NAME = ora11g)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = ora11g2)
      (ORACLE_HOME = C:\app\Administrator\product\11.1.0\db_1)
      (SID_NAME = ora11g2)
    )
  )



- create a text-based initialization parameter file for the auxiliary instance
that contains only one parameter: DB_NAME
in ORACLEHOME\database\initora11g2.ora
DB_NAME=ora11g2


- create Oracle SID service (in Windows)
oradim -new -sid ora11g2


- connect to the auxilary instance:
sqlplus /nolog
conn sys/ora11g@ora11g2 as sysoper
STARTUP NOMOUNT pfile=D:\oracle\product\11.1.0\db_1\database\InitOra11g2.ora



-- Start and Configure RMAN Before Duplication
# source database
RMAN>CONNECT TARGET SYS@ora11g
# duplicate database instance
RMAN>CONNECT AUXILIARY SYS@ora11g2



-- You may want to increase the parallelism setting of your source database
disk channels
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 ;


-- check parameters containing folder info:
col value format a35
col name format a30
select NAME, VALUE FROM v$parameter where upper(value) like '%D:\%';
```

```
-- Run the DUPLICATE command
-- duplicating a database with a different directory structure
--  ( CREATE REQUIRED DIRS)
DUPLICATE TARGET DATABASE TO ora11g2
FROM ACTIVE DATABASE
DB_FILE_NAME_CONVERT 'D:\oracle\oradata\ora11g','D:\oracle\oradata\ora11g2'
SPFILE
PARAMETER_VALUE_CONVERT 'D:\oracle\oradata\ora11g',
'D:\oracle\oradata\ora11g2'
SET MEMORY_TARGET '350M'
SET LOG_FILE_NAME_CONVERT 'D:\oracle\oradata\ora11g',
'D:\oracle\oradata\ora11g2'
SET audit_file_dest 'D:\oracle\admin\ora11g2\adump'
SET
CONTROL_FILES="D:\ORACLE\ORADATA\ORA11G2\CONTROL01.CTL","D:\ORACLE\ORADATA\ORA
11G2\CONTROL02.CTL";
```

Duplicating a Database without Recovery Catalog or Target Connection

- Oracle 11g R2

```
This is backup-based duplication

- Make Backups and Archived Logs Accessible to the Duplicate Instance:
-  if backup files location cannot be shared:
Create a new directory in the source host that has the same name as the
directory on the destination host that will contain the backups:
in source
mkdir C:\oracle\stagingbackups

in source
RUN {
 ALLOCATE CHANNEL c1 DEVICE TYPE DISK FORMAT 'C:\oracle\stagingbackups\%U';
 BACKUP AS COPY DATABASE;
 BACKUP AS COPY CURRENT CONTROLFILE FORMAT
'C:\oracle\stagingbackups\control01.ctl';
}
OR
RUN {
 ALLOCATE CHANNEL c1 DEVICE TYPE DISK FORMAT 'C:\oracle\stagingbackups\%U';
 BACKUP COPY OF DATABASE ;
 BACKUP AS COPY CURRENT CONTROLFILE FORMAT
'C:\oracle\stagingbackups\control01.ctl';
}

- Manually transfer the backups in the new directory on the source host to the
identically named directory on the destination host.

- in source:
-- check parameters containing folder info:
col value format a35
col name format a30
select NAME, VALUE FROM v$parameter where upper(value) like '%C:\%';
OR
select NAME, VALUE FROM v$parameter
where name in
```

```
('control_files','db_recovery_file_dest','audit_file_dest','diagnostic_dest');


- in destination:
-Create a password file in the destination server:
orapwd FILE=C:\oracle\oracledb11g\database\PWDora11g2.ora PASSWORD=ora11g
ENTRIES=10 ignorecase=n

- create a text-based initialization parameter file for the auxiliary instance
that contains only one parameter: DB_NAME
in ORACLEHOME\database\initora11g2.ora
DB_NAME=ora11g2


- create Oracle SID service (in Windows)
oradim -new -sid ora11g2


-configure Net connectivity to the auxiliary instance in the destination
server:
ORA11G2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = pc02)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ora11g2.pc02)
    )
  )


- configure the listner, if no listener already exists

- manually register the database in the listner.ora:

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = ora11g2.pc02)
      (ORACLE_HOME = C:\oracle\oracledb11g)
      (SID_NAME = ora11g2)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = PC02)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )



- connect to the auxilary instance:
set ORACLE_SID=ORA11G2
sqlplus /nolog
```

```
conn sys/ora11g@ora11g2 as sysoper
STARTUP NOMOUNT pfile=C:\oracle\oracledb11g\database\initora11g2.ora


-- Start and Configure RMAN Before Duplication
# duplicate database instance
set ORACLE_SID=ORA11G2
rman
CONNECT AUXILIARY SYS@ora11g2

-- You may want to increase the parallelism setting of your source database
disk channels
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 ;

-- Run the DUPLICATE command
-- duplicating a database with a different directory structure
--  ( CREATE REQUIRED DIRS)
DUPLICATE TARGET DATABASE TO ora11g2
BACKUP LOCATION 'C:\oracle\stagingbackups'
DB_FILE_NAME_CONVERT 'C:\oracle\oradata\ora11g','C:\oracle\oradata\ora11g2'
SPFILE
PARAMETER_VALUE_CONVERT 'C:\oracle\oradata\ora11g', 'C:\oracle\oradata\ora11g'
SET MEMORY_TARGET '620M'
SET LOG_FILE_NAME_CONVERT 'C:\oracle\oradata\ora11g',
'C:\oracle\oradata\ora11g2'
SET audit_file_dest 'C:\ORACLE\ADMIN\ORA11G2\ADUMP'
SET
CONTROL_FILES="C:\ORACLE\ORADATA\ORA11G2\CONTROL01.CTL","C:\ORACLE\ORADATA\ORA
11G2\CONTROL02.CTL";
```

Manually Duplicating a Database

```
-- it is assumed that the duplicate database will have different name
(ora11g2) from prod (ora11g)

1. Copy the prod database files to the target location.
- in prod
SHUTDOWN IMMEDIATE
COPY THE DATABASE DATA FILES (EXCLUDE THE TEMFILES) + redo log file + spfiles
CREATE PFILE='C:\TEMP\MYPFILE.TXT' FROM MEMORY;

Edit the generated PFILE to modify directory-based parameters like:
control_files
audit_file_dest
db_recovery_file_dest
diagnostic_dest

Change DB name:
db_domain
db_name


2. Prepare a text file for the creation of a control file for the new database
as follows:
- in prod
SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

```
in ORACLE_BASE\diag\rdbms\ora11g\ora11g\trace\ folder

Modify as required the parameters below, include them in the parameter file
and create their folders:
Note: if you define diagnostic_dest, you don't have to create its sub-
directories

-- The following are current System-scope REDO Log Archival related
-- parameters and can be included in the database initialization file.
--
-- LOG_ARCHIVE_DEST=''
-- LOG_ARCHIVE_DUPLEX_DEST=''
--
-- LOG_ARCHIVE_FORMAT=ARC%S_%R.%T
--
-- DB_UNIQUE_NAME="ora11g2"
--
-- LOG_ARCHIVE_CONFIG='SEND, RECEIVE, NODG_CONFIG'
-- LOG_ARCHIVE_MAX_PROCESSES=4
-- STANDBY_FILE_MANAGEMENT=MANUAL
-- STANDBY_ARCHIVE_DEST=C:\oracle\oracledb11g\RDBMS
-- FAL_CLIENT=''
-- FAL_SERVER=''
--
-- LOG_ARCHIVE_DEST_10='LOCATION=USE_DB_RECOVERY_FILE_DEST'
-- LOG_ARCHIVE_DEST_10='OPTIONAL REOPEN=300 NODELAY'
-- LOG_ARCHIVE_DEST_10='ARCH NOAFFIRM NOEXPEDITE NOVERIFY SYNC'
-- LOG_ARCHIVE_DEST_10='REGISTER NOALTERNATE NODEPENDENCY'
-- LOG_ARCHIVE_DEST_10='NOMAX_FAILURE NOQUOTA_SIZE NOQUOTA_USED
NODB_UNIQUE_NAME'
-- LOG_ARCHIVE_DEST_10='VALID_FOR=(PRIMARY_ROLE,ONLINE_LOGFILES)'
-- LOG_ARCHIVE_DEST_STATE_10=ENABLE
--
--

-- the current versions of all online logs are available
-- Note: the generated code will use "REUSE DATABASE .. NORESETLOGS"; change
it to "SET DATABASE.. RESETLOGS " if
--  you want to use a differnt db name
CREATE CONTROLFILE SET DATABASE "ORA11G2" RESETLOGS ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 'C:\ORACLE\ORADATA\ORA11G2\REDO01.LOG'  SIZE 50M,
  GROUP 2 'C:\ORACLE\ORADATA\ORA11G2\REDO02.LOG'  SIZE 50M,
  GROUP 3 'C:\ORACLE\ORADATA\ORA11G2\REDO03.LOG'  SIZE 50M
-- STANDBY LOGFILE
DATAFILE
  'C:\ORACLE\ORADATA\ORA11G2\SYSTEM01.DBF',
  'C:\ORACLE\ORADATA\ORA11G2\SYSAUX01.DBF',
  'C:\ORACLE\ORADATA\ORA11G2\UNDOTBS01.DBF',
  'C:\ORACLE\ORADATA\ORA11G2\USERS01.DBF',
  'C:\ORACLE\ORADATA\ORA11G2\FDA_TBS.DBF'
CHARACTER SET AR8MSWIN1256
;
```

```
3. in destination: Create a password file in the destination server:
orapwd FILE=C:\oracle\oracledb11g\database\PWDora11g2.ora PASSWORD=ora11g
ENTRIES=10 ignorecase=n


4. Create Oracle SID service (in Windows)
oradim -new -sid ora11g2


5. Configure the listner, if no listener already exists


6. Connect to the instance:
set ORACLE_SID=ORA11G2
sqlplus /nolog
conn / as sysdba
# use pfile from step 1
STARTUP NOMOUNT pfile=C:\oracle\oracledb11g\database\initora11g2.ora
sample of its contetns follows:

audit_file_dest='C:\ORACLE\ADMIN\ORA11G2\ADUMP'
audit_trail='DB'
compatible='11.1.0.0.0'
control_files='C:\ORACLE\ORADATA\ORA11G2\CONTROL01.CTL'
control_files='C:\ORACLE\ORADATA\ORA11G2\CONTROL02.CTL'
control_files='C:\ORACLE\ORADATA\ORA11G2\CONTROL03.CTL'
db_block_size=8192
db_domain='pc02'
db_name='ora11g2'
db_recovery_file_dest='C:\oracle\flash_recovery_area'
db_recovery_file_dest_size=4048M
diagnostic_dest='C:\ORACLE'
dispatchers='(PROTOCOL=TCP) (SERVICE=ora11gXDB)'
log_buffer=5654016 # log buffer update
memory_target=620M
open_cursors=400
optimizer_dynamic_sampling=2
optimizer_mode='ALL_ROWS'
plsql_warnings='DISABLE:ALL' # PL/SQL warnings at init.ora
processes=400
query_rewrite_enabled='TRUE'
remote_login_passwordfile='EXCLUSIVE'
result_cache_max_size=2112K
sessions=390
skip_unusable_indexes=TRUE
undo_tablespace='UNDOTBS1'
LOG_ARCHIVE_FORMAT=ARC%S_%R.%T
DB_UNIQUE_NAME="ora11g2"
LOG_ARCHIVE_DEST_10='LOCATION=USE_DB_RECOVERY_FILE_DEST'

7. Create the control file be executing the command in step 2

8. Execute:
ALTER DATABASE OPEN RESETLOGS;
```

```
8. Execute:
ALTER TABLESPACE TEMP ADD TEMPFILE 'C:\ORACLE\ORADATA\ORA11G2\TEMP01.DBF' size
250m autoextend on next 10m maxsize 4g;

9. Execute:
ALTER DATABASE RENAME GLOBAL_NAME TO ora11g2.pc02;
ALTER SYSTEM REGISTER;
```

Using Oracle Flashback Technology

Flashback Options

- Row Level:
  - Flashback Query
  - Flashback Versions Query
  - Flashback Transaction Query
  - Flashback Transaction Backout
- Table level:
  - Flashback Table
  - Flashback Drop
  - Flashback Data Archive (FDA)
- Database level:
  - Flashback Database

Preparing Your Database for Flashback

```
-- must be AUTO
show parameter UNDO_MANAGEMENT

-- set undo retension target
select value from v$parameter where upper(name)='UNDO_RETENTION';
alter system set UNDO_RETENTION = 2400;

-- get undo tbs name
show parameter UNDO_TABLESPACE

-- consider guaranteeing undo retention:
ALTER TABLESPACE UNDOTBS1 RETENTION GUARANTEE ;
```

Using Row Level Flashback Options

Flashback Query

```
-- if the user isn't the owner
GRANT FLASHBACK ON employees TO scott;
GRANT FLASHBACK ANY TABLE TO scott;

-- what was the salary at that time
SELECT employee_id, department_id, salary FROM employees
AS OF TIMESTAMP
TO_TIMESTAMP ('2008-11-07 11:30:00', 'YYYY-MM-DD HH:MI:SS')
WHERE employee_id=101;

select employee_id , salary from employees
  AS OF SCN 794532059
  where employee_id=101



/* using DBMS_FLASHBACK */
-- set the specified point in time in the past
select ... -- the output will be as of current time
EXECUTE DBMS_FLASHBACK.ENABLE_AT_TIME (TO_TIMESTAMP '11-DEC-2008 10:00:00',
'DD-MON-YYYY hh24:MI:SS');
select ... -- the output will be as of the set time
EXECUTE DBMS_FLASHBACK.DISABLE ();



-- to get current SCN
SELECT current_scn from V$DATABASE;
select DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER from dual;
```

Flashback Versions Query

- The VERSIONS clause cannot span DDL commands.
- Syntax:
  ```
  SELECT [pseudocolumns] . . . /* provide details about the row history */
  FROM . . . /* table name goes here */
  VERSIONS BETWEEN
  {SCN|TIMESTAMP {expr|MINVALUE} AND {expr|MAXVALUE}}
  [AS OF{SCN|TIMESTAMP expr}]
  WHERE [pseudocolumns . . . ] . . .
  ```
- Pseudo-Columns:
  ```
  VERSIONS_STARTSCN and VERSIONS_STARTTIME:
  VERSIONS_ENDSCN and VERSIONS_ENDTIME: if null, current or deleted
  VERSIONS_OPERATION: I,D,U
  VERSIONS_XID: trans ID
  ```

Note: The actual time might be up to three seconds earlier or later than the time you specify with a time stamp.

- Flashback Versions Query restrictions:
  - You can't apply the VERSIONS clause across DDL operations.
  - The query will ignore purely physical row changes as happen, for example, during a segment shrink operation.

```
SELECT versions_xid AS XID,
 versions_startscn AS START_SCN,
 versions_endscn AS END_SCN,
 versions_operation AS OPERATION,
 last_name,
salary
FROM EMPLOYEES
VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
WHERE employee_id = 101;

SET LINESIZE 100
COL START_TIME FORMAT A21
COL END_TIME FORMAT A21
SELECT versions_xid AS XID,
 VERSIONS_STARTTIME AS START_TIME,
 VERSIONS_ENDTIME AS END_TIME,
 VERSIONS_OPERATION AS OPERATION,
 last_name,
 salary
FROM EMPLOYEES
VERSIONS BETWEEN TIMESTAMP MINVALUE AND MAXVALUE
WHERE employee_id = 101;
```

Flashback Transaction Query

- It is actually a SELECT from FLASHBACK_TRANSACTION_QUERY

```
/* requirements */
--(1) highly recommended
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
--(2) required privs
grant SELECT ANY TRANSACTION to scott;

/* using it */
SELECT to_char(start_timestamp,'dd-Mon hh24:mi'), operation, undo_sql
FROM flashback_transaction_query
WHERE start_timestamp >= TO_TIMESTAMP ('2009-11-09 05:00:00', 'YYYY-MM-DD
HH:MI:SS')
  AND table_owner='HR' AND table_name='EMPLOYEES'
order by start_timestamp desc;

/* using flashback transaction with versions query */
SELECT xid, START_SCN , commit_scn COMMIT, OPERATION, logon_user, undo_sql
FROM flashback_transaction_query
WHERE xid = HEXTORAW('0600230050010000'); -- obtained from version query
```

Flashback Transaction (Backout)

- Target: to back-out a committed transaction and all dependent transactions while the database is still online.

- OEM provides an easier to manage interface.
- DBMS_FLASHBACK.TRANSACTION_BACKOUT parameters:

NUMBEROFXIDS	Number of transactions to be backed out
XIDS	array of transaction identifiers
OPTIONS	nocascade (default), cascade, nocascade_force, noconflict_only
TIMEHINT	time at the start of the transaction you are backing out
SCNHINT	SCN at the beginning of the transaction you are backing out

```
-- Setting Up for Flashback Transaction Backout
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;


-- required privs
grant EXECUTE on DBMS_FLASHBACK to hr;
grant SELECT ANY TRANSACTION to hr;


-- get the transaction ID
-- use v$ views or the flashback transaction query or log miner
-- following example from sql in cache:
select A.SQL_TEXT, B.XID
from   V$OPEN_CURSOR A, V$TRANSACTION B, V$SESSION C
where  A.SID=C.SID AND C.TADDR=B.ADDR
  and  A.SQL_TEXT LIKE '%delete%';


-- backout the trans
declare
 V_XID SYS.XID_ARRAY;
begin
 V_XID := SYS.XID_ARRAY('03001800BC0D0000');
 DBMS_FLASHBACK.TRANSACTION_BACKOUT( NUMTXNS => 1,
    XIDS=>V_XID, OPTIONS=>DBMS_FLASHBACK.CASCADE); -- children trans also
backed out
end;
/
-- to make it permanent: you must now COMMIT or ROLLBACK

-- Information about transaction Backouts done can be obtained from:
DBA_FLASHBACK_TXN_STATE: any transaction shown in this view is backed out.
DBA_FLASHBACK_TXN_REPORT: compensating status of all transactions in the
database
```

Using Table Level Flashback Options

Flashback Table

- ROW MOVEMENT must be enabled in advance and before flashback recovery time.
- Generates undo and redo data
- Syntax:

  `FLASHBACK TABLE [schema.]table [,[schema.]table] ... TO {SCN|TIMESTAMP} expr`
- By default, Oracle disables all relevant triggers and reenables them upon completing the table recovery.
- Several restrictions apply:
  - You can't flash back a table to a time preceding any DDL operation involving a change in table structure.
  - If the flashback operation involves multiple tables, all of the tables must be flashed back or none.

```
-- required privs
grant FLASHBACK ANY TABLE to hr;
grant FLASHBACK on hr.employees to hr;
-- SELECT, INSERT, DELETE, and ALTER privileges on the table to be flashed
back are required

-- required on the table
--  also enable it for the tables that reference it
ALTER TABLE emp ENABLE ROW MOVEMENT;

-- TAKE current SCN before flashback to go back to initial stat of table if
--   the flashback doesn't do good
SELECT current_scn from V$DATABASE;
select DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER from dual

FLASHBACK TABLE emp TO SCN 5759290864;

FLASHBACK TABLE emp TO TIMESTAMP TO_TIMESTAMP ('2008-01-30 07:00:00', 'YYYY-
MM-DD HH24:MI:SS');

FLASHBACK TABLE emp TO TIMESTAMP TO_TIMESTAMP ('2009-04-05 10:00:00', 'YYYY-
MM-DD HH24:MI:SS') ENABLE TRIGGERS;
```

Flashback Drop

- When table is dropped, its dependents' names will also renamed by system generated names.
- If a table is undropped, the cryptic system-generated names of the dependents remain.

```
-- Necessary Privileges ****
be the owner or
have the drop privileges (DROP TABLE or DROP ANY TABLE) on a table.
You must have the SELECT privilege and the FLASHBACK privilege on an object in
order to query that object in the Recycle Bin.
```

```
-- list dropped tables  ****
-- db level
SELECT owner, original_name, object_name,ts_name, droptime, CAN_UNDROP
FROM dba_recyclebin;

-- user level
SELECT * FROM RECYCLEBIN;
SHOW RECYCLEBIN


-- Enabling and Disabling the Recycle Bin ***
-- session level
ALTER SESSION SET recyclebin = OFF;
-- db level
-- it affects the parameter file
ALTER SYSTEM SET recyclebin = OFF;



-- retreive data from a dropped table
SELECT * FROM "BIN$xTMPjHZ6SG+1xnDIaR9E+g==$0";



-- restore table from drop
FLASHBACK TABLE table_name TO BEFORE DROP;
FLASHBACK TABLE "BIN$xTMPjHZ6SG+1xnDIaR9E+g==$0" TO BEFORE DROP;
FLASHBACK TABLE "BIN$xTMPjHZ6SG+1xnDIaR9E+g==$0" TO BEFORE DROP RENAME TO
NEW_PERSONS;
-- you can rename any dependent
SELECT INDEX_NAME FROM USER_INDEXES WHERE TABLE_NAME = 'JOB_HISTORY';
ALTER INDEX "BIN$DBo9UChtZSbgQFeMiAdCcQ==$0" RENAME TO JHIST_JOB_IX;

-- purge a table(s)
DROP TABLE table_name PURGE;
PURGE TABLE int_admin_emp;
PURGE TABLE BIN$jsleilx392mk2=293$0;
PURGE TABLESPACE users;
PURGE TABLESPACE users USER scott;
PURGE RECYCLEBIN;
-- purge all objects in the db
PURGE DBA_RECYCLEBIN;
```

Flashback Data Archive

- FDA is part of the "Oracle Total Recall" option in Oracle database 11g.

- Disabling flashback archiving for a table or dropping its flashback archive object will result in all the historical data for that table being lost. It also requires SYSDBA or FLASHBACK ARCHIVE ADMINISTER privilege.
- For a table with Flashback Archiving enabled, you cannot issue the following DDL commands: ALTER TABLE (except adding a column), DROP TABLE, RENAME TABLE and TRUNCATE TABLE.

```
/*  to obtain info about FDA */
-- information on all flashback archives contained in the database
SELECT * FROM DBA_FLASHBACK_ARCHIVE;

-- information on all tablespaces containing flashback archives
SELECT * FROM DBA_FLASHBACK_ARCHIVE_TS;

-- which flashback archive a given table is assigned to.
SELECT TABLE_NAME, OWNER_NAME, FLASHBACK_ARCHIVE_NAME FROM
DBA_FLASHBACK_ARCHIVE_TABLES;

-- history tables names created by Oracle:
select * from dba_FLASHBACK_ARCHIVE_TABLES;

/* Setting up the Data Flashback Archive */
--1) for system level FDA, FLASHBACK ARCHIVE ADMINISTER priv is required
SELECT * FROM DBA_SYS_PRIVS WHERE PRIVILEGE LIKE '%FLASHBACK ARC%'
-- for specific table:
GRANT FLASHBACK ARCHIVE ON hr_hist TO scott;



--2) Create FDA object
-- The statement in the example above may return ORA-55603: Invalid Flashback
Archive command error, if you try to create a flashback archive in a non-empty
tablespace. I figured out a workaround which is to put the tablespace name
between double quotations.
CREATE FLASHBACK ARCHIVE hr_hist  -- DEFAULT keyword may be used
TABLESPACE fda_archives -- mandatory (and it must be with ASSM)
QUOTA 5G -- optional in M,G,T,P
RETENTION 24 MONTH; -- mandatory (in YEAR, MONTH, DAY)



--3) Enable Flashback Data Archiving for existing or new tables
-- Create the table, using the default archive location.
CREATE TABLE my_table(..) FLASHBACK ARCHIVE;

-- Modify a table to use the default archive location
-- Note: if there is not default flashback archive, an error will be raised
ALTER TABLE my_table FLASHBACK ARCHIVE;

-- Create a table to use a non-default archivelocation
CREATE TABLE my_table (..) FLASHBACK ARCHIVE hr_arc;
```

```
-- Modify a table to use a non-default archive location.
ALTER TABLE my_table FLASHBACK ARCHIVE hr_arc;

-- Modify a table to stop (disable) archiving.
ALTER TABLE my_table NO FLASHBACK ARCHIVE;



/* Altering Flashback Archives */
-- make myflash the default flashback archive (as SYSDBA)
ALTER FLASHBACK ARCHIVE myflash SET DEFAULT;

-- add space to the flashback archive
ALTER FLASHBACK ARCHIVE myflash ADD TABLESPACE mytbs;

-- remove the tablespace from use by the flashback archive
-- (assign it to another tablespace first)
ALTER FLASHBACK ARCHIVE myflash REMOVE TABLESPACE mytbs;

-- change the quota for the archive
ALTER FLASHBACK ARCHIVE myflash MODIFY TABLESPACE mytbs QUOTA 10G;

-- undefined quota (make the space unlimited)
ALTER FLASHBACK ARCHIVE myflash MODIFY TABLESPACE mytbs;

-- change the archive retention time
ALTER FLASHBACK ARCHIVE myflash MODIFY RETENTION 2 YEAR;

-- purge all archived data
ALTER FLASHBACK ARCH`IVE myflash PURGE ALL;

-- purge data older than 2 days
ALTER FLASHBACK ARCHIVE MYFLASH
PURGE BEFORE TIMESTAMP( SYSTIMESTAMP – INTERVAL '2' DAY);

-- dropping FDA:
DROP FLASHBACK ARCHIVE myflash;

/* Using Oracle Flashback Data Archives */
SELECT LAST_NAME, SALARY FROM HR.EMPLOYEES
AS OF TIMESTAMP TO_TIMESTAMP ('2008-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS');
SELECT LAST_NAME, SALARY FROM HR.EMPLOYEES
AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '6' MONTH);

SELECT LAST_NAME, SALARY FROM HR.EMPLOYEES
VERSIONS BETWEEN TIMESTAMP
TO_TIMESTAMP ('2008-01-01 00:00:00','YYYY-MM-DD HH24:MI:SS') -- or MINVALUE
AND
TO_TIMESTAMP ('2008-01-01 15:00:00','YYYY-MM-DD HH24:MI:SS') -- or MAXVALUE
```

```
WHERE EMPLOYEE_ID = 200;

FLASHBACK TABLE employees TO TIMESTAMP (SYSTIMESTAMP – INTERVAL '6' MONTH);
```

Using Flashback Database

When to use Flashback Database

- To retrieve a dropped schema
- When a user error affects the entire database
- When you truncate a table in error
- When a batch job performs only partial changes

Flashback Database Considerations

- You cannot use Flashback Database when:
- The control file has been restored or re-created.
- A tablespace has been dropped.
- A data file has been reduced in size, for example by a shrink.

Using Flashback Database

- It is essential that you monitor the flash recovery area's size to ensure that you have sufficient space so as not to risk losing any of the Flashback Database logs.

```
/* Configuring Flashback Database */
-- Check that your database is in the archivelog mode:
ARCHIVE LOG LIST

-- in minutes ( 1440= 1 day)
ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET=1440;

-- in mount level
ALTER DATABASE FLASHBACK ON;
ALTER DATABASE OPEN;


/* Exempting a tablespace form db flashback */
ALTER TABLESPACE users FLASHBACK OFF;

-- to switch it on again:
ALTER TABLESPACE users FLASHBACK ON;


/* Disabling Flashback Database */
-- in mount level
-- all Flashback Database logs will be deleted
ALTER DATABASE FLASHBACK OFF;
```

```
/* Obtain info about Flashback Database */
-- to find out how far you can flash back your database
SELECT oldest_flashback_scn,
 to_char(oldest_flashback_time,'dd-Mon hh24:mi:ss') oldest_flashback_time
FROM v$flashback_database_log;

-- to estimate the space required by flashback db:
SELECT estimated_flashback_size/1024/1024 estimated_flashback_size,
 retention_target, flashback_size/1024/1024 flashback_size
FROM v$flashback_database_log;

-- last 24-hour stats data (each row is one-hour )
SELECT begin_time, end_time, flashback_data, db_data,
  redo_data, estimated_flashback_size AS EST_FB_SZE
FROM V$FLASHBACK_DATABASE_STAT
order by begin_time desc;



/* Flashback db in action */
-- flashback db to a past time
STARTUP MOUNT;
FLASHBACK DATABASE TO SCN 5964663;
FLASHBACK DATABASE TO SEQUENCE 12345;
FLASHBACK DATABASE TO TIMESTAMP(SYSDATE -1/24);
-- You can flash back to just before the last RESETLOGS operation by :
FLASHBACK DATABASE TO SCN 5964663 TO BEFORE RESETLOGS;
-- you can check the data before open resetlogs
ALTER DATABASE OPEN READ ONLY;

-- open resetlogs
ALTER DATABASE OPEN RESETLOGS;

-- to undo the results of the entire Flashback operation
RECOVER DATABASE;

/* Assissted commands */
-- current scn
SELECT current_scn FROM V$DATABASE;
```

Restore Points

- Guaranteed restore points use a separate logging mechanism from the Flashback logging used for a Flashback Database operation.
- However, if you use a guaranteed restore point and Flashback Database is enabled, Oracle won't delete any Flashback logs and thus FRA will eventually become full.
- Turning off Flashback Database if you're using guaranteed restore points.

- With guaranteed restore points, you can only flashback to exactly the restore point time.

```
-- obtain info about rp:
SELECT name, scn, storage_size, time, guarantee_flashback_database
FROM v$restore_point;


-- creating ordinary restore point
CREATE RESTORE POINT rp_test;
DROP RESTORE POINT rp_test;


-- creating a guaranteed restore point
CREATE RESTORE POINT test_guarantee GUARANTEE FLASHBACK DATABASE;


-- to use later the rp:
FLASHBACK DATABASE TO RESTORE POINT test_guarantee;


-- to know if flashback db status: if only guaranteed rp is enabled, it'll
return RESTORE POINT ONLY
SELECT flashback_on FROM v$database;
```

Part 4 Oracle Database Security

Oracle Database Security Management

- The main aspects of Oracle database security management:
  - **Authorization**: controlling access to data
  - **Authentication**: restricting access to legitimate users
  - **Auditing**: ensuring accountability on the part of the users
  - **Encryption**: safeguarding key data in the database
  - **Enterprise Security**: managing the security of the entire organizational information structure

Security Guidelines

- Apply Security Patches. http://technet.oracle.com/deploy/security/alerts.htm
- Enable "Automatic Secure Configuration" when creating a new database.
- Lock default users. Exceptions: SYS, SYSTEM, DBSNMP, SYSMAN, and MGMT_VIEW.
- Strict password aging and expiration policies.
- Do not use hard coded passwords in your script.
- If possible, avoid Operating System Authentication.
- Enable SYSDBA operations by setting AUDIT_SYS_OPERATIONS to TRUE. The audit records are written to the operating system's audit trail.
- Avoid granting ANY privileges and privileges with the ADMIN option.
- Whenver possible, use roles rather than granting privileges directly to users.
- Don't grant any unnecessary roles or privileges to PUBLIC.

```
select count(*) from DBA_TAB_PRIVS
 where GRANTEE='PUBLIC';
```

- Set the UMASK variable to 022 to the Oracle software OS user owner.
- Remove the SETUID on all Oracle files.
- Unless needed, remove mentions to EXTPROC in both the listener.ora file on the server and the tnsnames.ora file on the client. Then remove all EXTPROC executables from your $ORACLE_HOME/bin directory (usually extproc and xtproc0). If needed, refere to Document ID 175429.1 in metalink.
- Set a password to the listener. Secure the linstener.ora file.

  ADMIN_RESTRICTIONS=ON
- You may use Server-side Access Controls in the sqlnet.ora file as follows:

```
-- only the addresses in the list are allowed to make connections
tcp.validnode_checking = yes
tcp.invited_nodes = (server1.us.mycompany.com,172.14.16.152)
-- addresses in the list are excluded
tcp.excluded_nodes = (server1.us.mycompany.com,172.14.16.152)
```

- Disable remote OS authentication by setting the following parameter in the init.ora :

```
REMOTE_OS_AUTHENT=FALSE
```

- Consider the proper setting of the security-related parameters:

```
# password case sensitivity (11g)
select value from v$parameter where upper(name)='SEC_CASE_SENSITIVE_LOGON'
alter system set SEC_CASE_SENSITIVE_LOGON = TRUE;
# maximum number of authentication attempts that can be made by a
#  client on a connection to the server process. Default 10 (11g)
select value from v$parameter where
lower(name)='sec_max_failed_login_attempts'
alter system set sec_max_failed_login_attempts=10 scope=spfile;
```

- For application users:

- Grant privileges to users through roles rather than direct access.
- Consider using application enabled roles.
- The users can make data changes only through procedures not using direct DML statements.
- Consider preventing normal users from using SQL*Plus:

```
-- disable
INSERT INTO SYSTEM.PRODUCT_USER_PROFILE(PRODUCT,userid,attribute,char_value)
VALUES('SQL*Plus','TESTER','ROLES','TEST123');
-- enable
DELETE FROM product_user_profile  WHERE userid='TESTER'
 AND char_value = 'TEST123';
```

Managing Users

- Recommendation: create default tablespace to every user.
- For managing resources, it might be better using Resource Manager.
- User Profile parameters:
  - Resource limit paramters:
    - CONNECT_TIME: (in minutes)
    - IDLE_TIME
    - CPU_PER_CALL: CPU time used per each call
    - CPU_PER_SESSION
    - SESSIONS_PER_USER
    - LOGICAL_READS_PER_SESSION
    - LOGICAL_READS_PER_CALL
    - PRIVATE_SGA: applicable only to shared server architecture systems
    - COMPOSITE_LIMIT:
  - Password management parameters (see the following section):
    - FAILED_LOGIN_ATTEMPTS
    - PASSWORD_LIFE_TIME
    - PASSWORD_GRACE_TIME
    - PASSWORD_LOCK_TIME
    - PASSWORD_REUSE_TIME
    - PASSWORD_REUSE_MAX
    - PASSWORD_VERIFY_FUNCTION

```
-- creating user
CREATE USER user1 IDENTIFIED BY urs1754
TEMPORARY TABLESPACE TEMPTBS01
DEFAULT TABLESPACE user1ts
QUOTA 500M ON user1ts
PROFILE 'SALES_RPOF';

GRANT CREATE SESSION TO salapati;
ALTER USER user1 QUOTA 100M ON user1ts;
GRANT UNLIMITED TABLESPACE TO user1;

SELECT tablespace_name, username, bytes FROM DBA_TS_QUOTAS
 WHERE username='USER1';

-- dropping user
DROP USER user1;
DROP USER user1 cascade;
REVOKE CREATE SESSION FROM user1;
```

```
-- Creating and Using User Profiles
CREATE PROFILE SALES_PROF
 LIMIT
 connect_time 120
 failed_login_attempts 3
 idle_time 60
 sessions_per_user 2;

ALTER PROFILE test
 LIMIT
 sessions_per_user 4
 failed_login_attempts 4;

-- to make Oracle enforce the resource limits in
--  the profile, if they are used
ALTER SYSTEM SET resource_limit=true;

ALTER USER salapati PROFILE SALES_PROF;

SELECT profile FROM dba_users
WHERE username = 'USER1';

SELECT DISTINCT resource_name, limit
FROM dba_profiles
WHERE profile='DEFAULT';


-- Password Management Function
-- utlpwdmg.sql script creates verify_function_11g
ALTER PROFILE DEFAULT
LIMIT
 PASSWORD_LIFE_TIME 180
 PASSWORD_GRACE_TIME 7
 PASSWORD_REUSE_TIME UNLIMITED
 PASSWORD_REUSE_MAX UNLIMITED
 FAILED_LOGIN_ATTEMPTS 10
 PASSWORD_LOCK_TIME 1
 PASSWORD_VERIFY_FUNCTION verify_function_11G;

-- Dropping a User Profile
DROP PROFILE test CASCADE;
```

Database Authentication

Managin Passwords

```
select value from v$parameter where name='sec_case_sensitive_logon';
-- dynamic
alter system set sec_case_sensitive_logon=false ;

SELECT username, password, password_versions
FROM dba_users
order by 1;

/*  making a password expired */
ALTER USER hr IDENTIFIED BY hr PASSWORD EXPIRE;
ALTER PROFILE test_profile
 LIMIT PASSWORD_LIFE_TIME 30; -- in days (refere to profile section)
ALTER USER hr PROFILE test_profile;

/* password file */
select value from v$parameter where upper(name)='REMOTE_LOGIN_PASSWORDFILE';
SELECT * FROM v$pwfile_users;

orapwd FILE=testpwd PASSWORD=remorse1 ENTRIES=20
```

External (OS) Authentication

- Following are the steps to setting up OS Authentication on UNIX/Linux and Windows platforms.

```
# to enable external authentication over the net
REMOTE_OS_AUTHENT=TRUE



# create the OS user
useradd ahmedb
passwd ahmedb

# set the parameter OS_AUTHENT_PREFIX
SHOW PARAMETER os_authent_prefix
alter system set os_authent_prefix='ops$' scope=spfile;
alter system set OS_AUTHENT_PREFIX = '' scope=spfile;

# in DB: create the DB user with the prefix
# in Unix
CREATE USER ops$ahmedb IDENTIFIED EXTERNALLY;
GRANT CONNECT TO ops$ahmedb;
# in Windows
```

```
CREATE USER "OPS$MYDOMAIN.COM\AHMEDB" IDENTIFIED EXTERNALLY;
GRANT CONNECT TO "OPS$MYDOMAIN.COM\AHMEDB";

# in Windows:
# set the following in the file "%ORACLE_HOME%\network\admin\sqlnet.ora":
SQLNET.AUTHENTICATION_SERVICES= (NTS)

# test:
su - ahmedb
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/db_1
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=ora11gr2
sqlplus /
```

Proxy Authentication

```
-- to authorize connections by a database user logging on from
-- a middle-tier node, using password authentication.
ALTER USER user1
GRANT CONNECT THROUGH appserv
AUTHENTICATED USING PASSWORD;

--
ALTER USER user1 GRANT CONNECT THROUGH appserv;
```

Logging In As a Different User

- If you want to log on as a different user and you do not know his password.

```
-- 1) obtain his encrypted password
SELECT 'alter user tester identified by values '||password||';'
 FROM user$
 WHERE username='HR';
-- 2) set a new password to him
ALTER USER hr IDENTIFIED BY newpassword;
-- 3) log on using the new password and execute your commands
-- 4) reset the user password back to the original one (note the qout)
ALTER USER tester IDENTIFIED BY VALUES '1825ACAA229030F1';
```

Killing User Sessions from OS

```
-- on Unix
-- 1. obtain process#
SELECT process, sid, serial# FROM v$session WHERE username='&user';
-- 2. kill the process
kill -9 345678
```

```
-- on Windows
-- 1. obain thread #
SELECT sid, spid as thread, osuser, s.program
FROM v$process p, v$session s
WHERE p.addr = s.paddr;
-- kill the process using orakill utiliy (orakill DBsid thread#)
orakill MyDB 6200
```

Controlling Database Access

System and Object Privileges

```
/* system privs */
GRANT CREATE SESSION TO hr;
GRANT CREATE SESSION TO hr WITH ADMIN OPTION;
GRANT ANY OBJECT to hr;
-- only SELECT ANY DICTIONARY is NOT included
GRANT ALL PRIVILEGES TO ourdba;
REVOKE DELETE ANY TABLE FROM ourdba;

-- dictionary views become accessible
grant SELECT_CATALOG_ROLE to ourdev;


/* object privs */
GRANT DELETE ON bonuses TO hr WITH GRANT OPTION;
GRANT UPDATE (product_id) ON sales01 TO hr;
GRANT SELECT, UPDATE ON emp_view TO PUBLIC;
GRANT SELECT ON oe.customers_seq TO hr;
GRANT ALL ON EMPLOYEES TO hr;
GRANT EXECUTE ON employee_pkg TO hr;
GRANT QUERY REWRITE TO hr;
GRANT READ ON DIRECTORY bfile_dir TO hr;

-- incorrect
REVOKE UPDATE (hostname) ON ods_process FROM hr;
-- correct
REVOKE UPDATE ON ods_process FROM hr;
```

Invoker Rights and Definer Rights

```
CREATE OR REPLACE PROCEDURE delete_emp (p_emp_id number)
 AUTHID CURRENT_USER
IS
BEGIN
 DELETE FROM emp WHERE
 emp_id = p_emp_id;
 COMMIT;
END;
/
```

Roles

- Predefined roles:

- CONNECT
- RESOURCE: CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
- DBA
- EXP_FULL_DATABASE
- IMP_FULL_DATABASE
- RECOVERY_CATALOG_OWNER

```
-- create role
CREATE ROLE new_dba;

/* Role Authorization */
-- (1) Database authorization
CREATE ROLE clerk IDENTIFIED BY password;
-- (2) Database authorization with a PL/SQL package
-- the role is enabled by a hr.admin package:
CREATE ROLE admin_role IDENTIFIED USING hr.admin;
-- (3) Externally
CREATE ROLE accts_rec IDENTIFIED EXTERNALLY;
-- (4) Globally: enabled by an enterprise directory service
CREATE ROLE supervisor IDENTIFIED GLOBALLY;

/* Disabling and Enabling a Role */
-- disable
INSERT INTO SYSTEM.PRODUCT_USER_PROFILE(PRODUCT,userid,attribute,char_value)
VALUES('SQL*Plus','TESTER','ROLES','TEST123');
-- enable
DELETE FROM product_user_profile  WHERE userid='TESTER'
 AND char_value = 'TEST123';

-- Dropping a Role
DROP ROLE admin_user;
```

Users, Roles, and Privileges Views

DBA_USERS	Provides information about users
DBA_ROLES	Shows all the roles in the database
DBA_COL_PRIVS	Shows column-level object grants
DBA_ROLE_PRIVS	Shows users and their roles
DBA_SYS_PRIVS	Shows users who have been granted system privileges
DBA_TAB_PRIVS	Shows users and their privileges on tables
ROLE_ROLE_PRIVS	Shows roles granted to roles
ROLE_SYS_PRIVS	Shows system privileges granted to roles
ROLE_TAB_PRIVS	Shows table privileges granted to roles
SESSION_PRIVS	Shows privileges currently enabled for the current session
SESSION_ROLES	Shows roles currently enabled for the current session

```
-- list of everything granted to a user:
SELECT 'GRANT ' || PRIVILEGE||' to "'||grantee||'";' FROM dba_sys_privs WHERE
```

```
grantee='FINANCE'
union all
SELECT 'GRANT ' || GRANTED_ROLE||' to "'||grantee||'";' FROM dba_role_privs
WHERE grantee='FINANCE'
union all
SELECT 'GRANT ' || PRIVILEGE||' to "'||grantee||'";' FROM dba_tab_privs WHERE
grantee='FINANCE';
```

Fine-Grained Data Access (Virtual Private Database VPD)

- Fine-grained security within the database is implemented by DBMS_RLS
- Fine-grained access control policy: attached to DB object. It can be applied to SELECT, INSERT, UPDATE, INDEX, and DELETE statements.
- SYS user is not affected by FGAC.
- Common Predefined Attributes in the USERENV

instance	Instance ID
entryID	Auditing entry identifier
current_user	Name of the user who started the session
session_user	Database username by which the current user is authenticated
db_name	Name of the database
host	Name of the machine on which the database is running
os_user	Operating system account name
terminal	Client terminal through which the database is being accessed
ip_address IP	address of the client machine
external_name	External name of the database user

```
/* Using Application Context */
-- using pre-defined context
SELECT sys_context ('USERENV', 'OS_USER') FROM DUAL;
SELECT first_name,last_name,employee_id FROM employees
 WHERE UPPER(last_name)=sys_context('USERENV', 'SESSION_USER');

-- using user-defined application context: var=val pair per session
CONNECT system/system_passwd;
GRANT CREATE ANY CONTEXT TO hr;
CONNECT hr/hr;
CREATE CONTEXT employee_info USING hr.context;
CREATE OR REPLACE PACKAGE hr_context AS
 PROCEDURE select_emp_no ;
END;
/
CREATE OR REPLACE PACKAGE BODY hr_context as
 PROCEDURE select_emp_no IS
  empnum number;
 BEGIN
  SELECT employee_id INTO empnum FROM employees WHERE
  UPPER(last_name) = sys_context('USERENV', 'SESSION_USER');
```

```
  DBMS_SESSION.SET_CONTEXT('EMPLOYEE_INFO', 'EMP_NUM', EMPNUM);
 END select_emp_no;
END;
/
-- set application context: upon login trigger
CREATE OR REPLACE TRIGGER hr.security_context
 AFTER LOGON ON DATABASE
BEGIN
 hr_context.select_emp_no;
END;
/

/* using fine-grained access control */
-- here are the steps
--(1) define the policy function: which will generate the predicates
-- the function must adhere to the following syntax:
-- FUNCTION policy_function (object_schema IN VARCHAR2, object_name VARCHAR2)
--   RETURN VARCHAR2 returning max of 2000 Bytes
CREATE OR REPLACE PACKAGE hr_security AS
 FUNCTION empnum_sec (A1 VARCHAR2, A2 VARCHAR2)
 RETURN varchar2;
END;
/
CREATE OR REPLACE PACKAGE BODY hr_security AS
FUNCTION empnum_sec (A1 VARCHAR2, A2 VARCHAR2)
 RETURN varchar2
IS
 d_predicate varchar2 (2000);
BEGIN
 d_predicate:= 'employee_id =
 SYS_CONTEXT("EMPLOYEE_INFO","EMP_NUM")';
 RETURN d_predicate;
END empnum_sec;
END hr_security;
/
-- to make all db user being able to use it
GRANT EXECUTE ON hr_security TO public;

--(2) create security policy RLS=row-level security
-- POLICY_TYPE parameter in DBMS_RLS.ADD_POLICY takes:
--  DBMS_RLS.DYNAMIC (default)
--  DBMS_RLS.STATIC
--  DBMS_RLS.SHARED_STATIC
--  DBMS_RLS.CONTEXT_SENSITIVE
--  DBMS_RLS.SHARED_CONTEXT_SENSITIVE
BEGIN
DBMS_RLS.ADD_POLICY (OBJECT_SCHEMA=>'HR',
 OBJECT_NAME=>'EMPLOYEES',
 POLICY_NAME=>'MANAGER_POLICY',
```

```
 FUNCTION_SCHEMA=>'HR',
 POLICY_FUNCTION=>'hr_security.empnum_sec', -- func generates the predicates
 STATEMENT_TYPES='SELECT' );
END;
/
SELECT object_name, policy_name, sel, ins, upd, del, enable
FROM all_policies;

-- Column-Level VPD
-- You can apply column-level VPD to a table or a view
BEGIN
DBMS_RLS.ADD_POLICY (OBJECT_SCHEMA=>'HR',
 OBJECT_NAME=>'EMPLOYEES',
 POLICY_NAME=>'MANAGER_POLICY',
 FUNCTION_SCHEMA=>'HR',
 POLICY_FUNCTION=>'hr_security.empnum_sec',
 STATEMENT_TYPE='INSERT,UPDATE',
 SEC_RELEVANT_COLS=>'salary,commission')
END;
/

/* Exempting from Access Policy  */
GRANT EXEMPT ACCESS POLICY to hr;
```

Auditing Database

Standard Auditing

- Set AUDIT_TRAIL to: NONE (default), OS, DB (SYS.AUD$), DB_EXTENDED (SYS.AUD$ + the columns SQLBIND and SQLTEXT CLOB), XML (in OS), 'XML, EXTENDED'.
- If you set AUDIT_TRAIL to DB, change the tablespace of SYS.AUD$ from SYSTEM.
- Audited info:
    - Operating system login
    - Database username
    - Terminal and session identifiers
    - Operation performed or attempted
    - Date and time stamp
    - SQL text that triggered the auditing

```
select value from v$parameter where name='audit_trail';
-- if not defined: $ORACLE_HOME/rdbms/audit/
select value from v$parameter where name='audit_file_dest';

alter system set audit_trail=db_extended  scope=spfile;
ALTER SYSTEM SET audit_trail = xml,extended SCOPE=SPFILE ;

-- AUDIT focused by
--  DB/user
--  success/failure and
--  grouped by session or access
audit session ;
AUDIT SESSION BY hr;
AUDIT select table BY hr BY SESSION;
AUDIT DELETE ANY TABLE BY hr WHENEVER NOT SUCCESSFUL;
AUDIT UPDATE ANY TABLE;
AUDIT SELECT,INSERT,UPDATE,DELETE ON employees BY ACCESS WHENEVER SUCCESSFUL;
AUDIT ALL PRIVILEGES;

/* turn audit off */
NOAUDIT select table BY hr;
NOAUDIT ALL; /* turns off all statement auditing */
NOAUDIT ALL PRIVILEGES; /* turns off all privilege auditing */
NOAUDIT ALL ON DEFAULT; /* turns off all object auditing */

/* obtain info on audit */
select * from  DBA_STMT_AUDIT_OPTS;
select OS_USERNAME, USERNAME, USERHOST, ACTION_NAME,
to_char(LOGOFF_TIME,'dd-mm-yy hh24:mi:ss') LOGOFF, SESSION_CPU
from  DBA_AUDIT_SESSION;

select USERNAME, USERHOST, ACTION_NAME, OBJ_NAME
from  DBA_AUDIT_OBJECT
```

```
ORDER BY USERNAME;


/* flushing db audit */
CONN / AS SYSDBA
DELETE FROM SYS.AUD$;
```

Customizing Database Auditing with Triggers

```
/* The following procedure generate code required to write an auditing
    trigger
  Usage: execute the procedure and pass the table name. Then select from VW
   order by I to view the code */
CREATE TABLE VW ( I NUMBER, X VARCHAR2(4000));
SEQUENCE SEQ_I ;

CREATE PROCEDURE CREATE_AUDIT2 ( P_TABLE VARCHAR2)
IS
 V_NCOL_LIST VARCHAR2(4000) ;
 V_OCOL_LIST VARCHAR2(4000) ;
 V_NCOL_LIST2 VARCHAR2(4000) ;
 V_OCOL_LIST2 VARCHAR2(4000) ;
 V_OUTPUT VARCHAR2(4000);

 PROCEDURE INSERT_VW( V VARCHAR2) IS
 BEGIN
  INSERT INTO VW  VALUES(SEQ_I.NEXTVAL, V );
 END;

BEGIN
 -- ** generate the target Audit Table
 -- create sequence
 INSERT_VW('CREATE SEQUENCE SEQ_'|| P_TABLE || '_AUDIT ;');
 INSERT_VW('CREATE TABLE '|| P_TABLE || '_AUDIT (');
 INSERT_VW('ID NUMBER CONSTRAINT '|| P_TABLE || '_AUDIT_ID PRIMARY KEY,');
 -- N Columns
 FOR R IN ( SELECT
TABLE_NAME,COLUMN_NAME,DATA_TYPE,DATA_LENGTH,DATA_PRECISION,DATA_SCALE FROM
USER_TAB_COLUMNS WHERE TABLE_NAME = UPPER(P_TABLE) ) LOOP
  IF R.DATA_TYPE = 'NUMBER' AND R.DATA_PRECISION IS NULL THEN
    INSERT_VW( 'N' || R.COLUMN_NAME || ' NUMBER ,');
        V_NCOL_LIST := V_NCOL_LIST ||  'N' || R.COLUMN_NAME || ',';
        V_NCOL_LIST2 := V_NCOL_LIST2 || ' :NEW.' || R.COLUMN_NAME || ',';
  ELSIF R.DATA_TYPE = 'NUMBER' AND R.DATA_PRECISION IS NOT NULL AND
R.DATA_SCALE = 0 THEN
    INSERT_VW( 'N' || R.COLUMN_NAME || ' NUMBER('|| R.DATA_PRECISION ||'),');
        V_NCOL_LIST := V_NCOL_LIST ||  'N' || R.COLUMN_NAME || ',';
        V_NCOL_LIST2 := V_NCOL_LIST2 || ' :NEW.' || R.COLUMN_NAME || ',';
```

```
  ELSIF R.DATA_TYPE = 'NUMBER' AND R.DATA_PRECISION IS NOT NULL AND
R.DATA_SCALE <> 0 THEN
    INSERT_VW( 'N' || R.COLUMN_NAME || ' NUMBER('|| R.DATA_PRECISION ||','||
R.DATA_SCALE ||'),');
        V_NCOL_LIST := V_NCOL_LIST ||  'N' || R.COLUMN_NAME || ',';
        V_NCOL_LIST2 := V_NCOL_LIST2 || ' :NEW.' || R.COLUMN_NAME || ',';
  ELSIF R.DATA_TYPE = 'VARCHAR2' THEN
    INSERT_VW( 'N' || R.COLUMN_NAME || ' VARCHAR2('|| R.DATA_LENGTH ||'),');
        V_NCOL_LIST := V_NCOL_LIST ||  'N' || R.COLUMN_NAME || ',';
        V_NCOL_LIST2 := V_NCOL_LIST2 || ' :NEW.' || R.COLUMN_NAME || ',';
  ELSIF R.DATA_TYPE = 'DATE' THEN
    INSERT_VW( 'N' || R.COLUMN_NAME || ' DATE,');
        V_NCOL_LIST := V_NCOL_LIST ||  'N' || R.COLUMN_NAME || ',';
        V_NCOL_LIST2 := V_NCOL_LIST2 || ' :NEW.' || R.COLUMN_NAME || ',';
  END IF;
 END LOOP;
 -- O columns
 FOR R IN ( SELECT
TABLE_NAME,COLUMN_NAME,DATA_TYPE,DATA_LENGTH,DATA_PRECISION,DATA_SCALE FROM
USER_TAB_COLUMNS WHERE TABLE_NAME = UPPER(P_TABLE) ) LOOP
  IF R.DATA_TYPE = 'NUMBER' AND R.DATA_PRECISION IS NULL THEN
    INSERT_VW( 'O' || R.COLUMN_NAME || ' NUMBER ,');
        V_OCOL_LIST := V_OCOL_LIST ||  'O' || R.COLUMN_NAME || ',';
        V_OCOL_LIST2 := V_OCOL_LIST2 || ' :OLD.' || R.COLUMN_NAME || ',';
  ELSIF R.DATA_TYPE = 'NUMBER' AND R.DATA_PRECISION IS NOT NULL AND
R.DATA_SCALE = 0 THEN
    INSERT_VW( 'O' || R.COLUMN_NAME || ' NUMBER('|| R.DATA_PRECISION ||'),');
        V_OCOL_LIST := V_OCOL_LIST ||  'O' || R.COLUMN_NAME || ',';
        V_OCOL_LIST2 := V_OCOL_LIST2 || ' :OLD.' || R.COLUMN_NAME || ',';
  ELSIF R.DATA_TYPE = 'NUMBER' AND R.DATA_PRECISION IS NOT NULL AND
R.DATA_SCALE <> 0 THEN
    INSERT_VW( 'O' || R.COLUMN_NAME || ' NUMBER('|| R.DATA_PRECISION ||','||
R.DATA_SCALE ||'),');
        V_OCOL_LIST := V_OCOL_LIST ||  'O' || R.COLUMN_NAME || ',';
        V_OCOL_LIST2 := V_OCOL_LIST2 || ' :OLD.' || R.COLUMN_NAME || ',';
  ELSIF R.DATA_TYPE = 'VARCHAR2' THEN
    INSERT_VW( 'O' || R.COLUMN_NAME || ' VARCHAR2('|| R.DATA_LENGTH ||'),');
        V_OCOL_LIST := V_OCOL_LIST ||  'O' || R.COLUMN_NAME || ',';
        V_OCOL_LIST2 := V_OCOL_LIST2 || ' :OLD.' || R.COLUMN_NAME || ',';
  ELSIF R.DATA_TYPE = 'DATE' THEN
    INSERT_VW( 'O' || R.COLUMN_NAME || ' DATE,');
        V_OCOL_LIST := V_OCOL_LIST ||  'O' || R.COLUMN_NAME || ',';
        V_OCOL_LIST2 := V_OCOL_LIST2 || ' :OLD.' || R.COLUMN_NAME || ',';
  END IF;
 END LOOP;
 INSERT_VW( 'ACTION VARCHAR2(6),');
 INSERT_VW( 'ACTION_TIME DATE,');
 INSERT_VW( 'HOST_NAME    VARCHAR2(300),');
 INSERT_VW( 'OS_USERNAME  VARCHAR2(250));');
 -- generate the Auditing Trigger
```

```
 INSERT_VW('CREATE OR REPLACE TRIGGER TRG_' || P_TABLE || '_AUDIT');
 INSERT_VW('AFTER INSERT OR UPDATE OR DELETE ON '|| P_TABLE);
 INSERT_VW('FOR EACH ROW');
 INSERT_VW('BEGIN');
 INSERT_VW('IF INSERTING THEN');
 INSERT_VW('INSERT INTO ' || P_TABLE || '_AUDIT(ID,');
 INSERT_VW(V_NCOL_LIST);
 INSERT_VW('ACTION,ACTION_TIME,HOST_NAME,OS_USERNAME)');
 INSERT_VW('VALUES');
 INSERT_VW('(SEQ_'|| P_TABLE || '_AUDIT.NEXTVAL,');
 INSERT_VW( V_NCOL_LIST2);
 INSERT_VW( '''INSERT'',sysdate,
SYS_CONTEXT(''USERENV'',''HOST''),SYS_CONTEXT(''USERENV'',''OS_USER''));');
 INSERT_VW('END IF;');
 INSERT_VW('IF DELETING THEN');
 INSERT_VW('INSERT INTO ' || P_TABLE || '_AUDIT(ID,');
 INSERT_VW(V_OCOL_LIST);
 INSERT_VW('ACTION,ACTION_TIME,HOST_NAME,OS_USERNAME)');
 INSERT_VW('VALUES');
 INSERT_VW('(SEQ_'|| P_TABLE || '_AUDIT.NEXTVAL,');
 INSERT_VW( V_OCOL_LIST2);
 INSERT_VW( '''DELETE'',
SYSDATE,SYS_CONTEXT(''USERENV'',''HOST''),SYS_CONTEXT(''USERENV'',''OS_USER'')
);');
 INSERT_VW('END IF;');
 INSERT_VW('IF UPDATING THEN');
 INSERT_VW('INSERT INTO ' || P_TABLE || '_AUDIT(ID,');
 INSERT_VW(V_NCOL_LIST );
 INSERT_VW( V_OCOL_LIST);
 INSERT_VW('ACTION,ACTION_TIME,HOST_NAME,OS_USERNAME)');
 INSERT_VW('VALUES');
 INSERT_VW('(SEQ_'|| P_TABLE || '_AUDIT.NEXTVAL,');
 INSERT_VW( V_NCOL_LIST2 );
 INSERT_VW( V_OCOL_LIST2 );
 INSERT_VW( '''UPDATE'',
SYSDATE,SYS_CONTEXT(''USERENV'',''HOST''),SYS_CONTEXT(''USERENV'',''OS_USER'')
);');
 INSERT_VW('END IF;');
 INSERT_VW('END;');
 COMMIT;
END CREATE_AUDIT2;
```

Auditing the Database Using System Trigger

- System-Level trigger types:
  - Database startup
  - Logon and Logoff
  - DDL
  - Server error

```
/* Required privs */
grant ADMINISTER DATABASE TRIGGER to user1;

/* Obtain Info about DB triggers */
-- list the db triggers
SELECT a.obj#, a.sys_evts, b.name
FROM trigger$ a,obj$ b
WHERE a.sys_evts > 0
AND a.obj#=b.obj#
AND baseobject = 0;

/* Examples */

/* If db cannot start because of an error in the AFTER STARTUP trigger */
set linesize 150
col NAME format a30
col VALUE format a20
col DESCRIPTION format a60
SELECT x.ksppinm NAME, y.ksppstvl VALUE, ksppdesc DESCRIPTION
FROM x$ksppi x, x$ksppcv y
WHERE x.inst_id = userenv('Instance')
AND y.inst_id = userenv('Instance')
AND x.indx = y.indx
AND x.ksppinm = '_system_trig_enabled';
-- enable or disable db triggers
ALTER SYSTEM SET "_system_trig_enabled" = TRUE SCOPE=BOTH;

-- Log On Log Off trigger
CREATE OR REPLACE TRIGGER logon_audit
AFTER LOGON ON DATABASE
BEGIN
  INSERT INTO connection_audit
  (login_date, user_name)
  VALUES
  (SYSDATE, USER);
END logon_audit;
/
CREATE OR REPLACE TRIGGER logoff_audit_trig
 AFTER LOGOFF
 ON DATABASE
BEGIN
 INSERT INTO logon_audit
 VALUES
 (user,
 sys_context('userenv', 'sessionid'),
 null,
 sysdate,
 sys_context('userenv', 'host'));
END;
/

-- trigger to trap unsuccessful logons
/*
other errors that could be trapped include:
ORA-01004 - default username feature not supported
ORA-01005 - null password given
ORA-01035 - Oracle only available to users with restricted session priv
ORA-01045 - create session privilege not granted
*/
CREATE OR REPLACE TRIGGER logon_failures
AFTER SERVERERROR
ON DATABASE
BEGIN
```

```
  IF (IS_SERVERERROR(1017)) THEN
    INSERT INTO connection_audit
    (login_date, user_name)
    VALUES
    (SYSDATE, 'ORA-1017');
  END IF;
END logon_failures;
/

-- DDL triggers
/* BEFORE / AFTER ALTER
BEFORE / AFTER ANALYZE
BEFORE / AFTER ASSOCIATE STATISTICS
BEFORE / AFTER AUDIT
BEFORE / AFTER COMMENT
BEFORE / AFTER CREATE
BEFORE / AFTER DDL
BEFORE / AFTER DISASSOCIATE STATISTICS
BEFORE / AFTER DROP
BEFORE / AFTER GRANT
BEFORE / AFTER NOAUDIT
BEFORE / AFTER RENAME
BEFORE / AFTER REVOKE
BEFORE / AFTER TRUNCATE
AFTER SUSPEND */
CREATE OR REPLACE TRIGGER  ddl_log_trig
 AFTER DDL ON DATABASE
BEGIN
 INSERT INTO ddl_log
 (username, change_date, object_type, object_owner, database, event_name )
 VALUES
 (ORA_LOGIN_USER, sysdate, ora_dict_obj_type, ora_dict_obj_owner,
 ora_database_name, ora_sysevent)
END;
/
-- Disable granting privileges to PUBLIC
CREATE OR REPLACE TRIGGER ddl_trig
BEFORE GRANT
ON DATABASE
DECLARE
 g_list DBMS_STANDARD.ORA_NAME_LIST_T;
 n      PLS_INTEGER;
BEGIN
  n := ORA_GRANTEE(g_list);
  FOR i IN 1..n LOOP
    IF g_list(i) = 'PUBLIC' THEN
      RAISE_APPLICATION_ERROR(-20997,'Public Grants Not Allowed');
    END IF;
  END LOOP;
END;
/

-- System Errors
CREATE TABLE servererror_log (
error_datetime  TIMESTAMP,
error_user      VARCHAR2(30),
db_name         VARCHAR2(9),
error_stack     VARCHAR2(2000),
captured_sql    VARCHAR2(1000));

CREATE OR REPLACE TRIGGER log_server_errors
AFTER SERVERERROR
ON DATABASE
```

```
DECLARE
 captured_sql VARCHAR2(1000);
BEGIN
  SELECT q.sql_text
  INTO captured_sql
  FROM gv$sql q, gv$sql_cursor c, gv$session s
  WHERE s.audsid = audsid
  AND s.prev_sql_addr = q.address
  AND q.address = c.parent_handle;

  INSERT INTO servererror_log
  (error_datetime, error_user, db_name,
   error_stack, captured_sql)
  VALUES
  (systimestamp, sys.login_user, sys.database_name,
  dbms_utility.format_error_stack, captured_sql);
END log_server_errors;
/
```

Using Fine Grained Auditing

```
-----
-- Show all currently active FGA Policies in the database
-----
COL object_schema       FORMAT A10      HEADING 'Object|Schema'
COL object_name     FORMAT A20      HEADING 'Object Name' WRAP
COL policy_name     FORMAT A16      HEADING 'Policy Name' WRAP
COL policy_text     FORMAT A24      HEADING 'Policy Text' WRAP
COL policy_column   FORMAT A16      HEADING 'Policy Column' WRAP
COL enabled         FORMAT A05      HEADING 'On?'
COL siud_options    FORMAT A04      HEADING 'SIUD|Set'
SELECT
     policy_name
    ,policy_text
    ,policy_column
    ,enabled
    ,object_schema
    ,object_name
    ,DECODE(sel,'YES','Y','N') || DECODE(ins,'YES','Y','N')||
     DECODE(upd,'YES','Y','N') || DECODE(del,'YES','Y','N') siud_options
  FROM dba_audit_policies;

-- required priv
grant execute on DBMS_FGA to hr;

-----
-- Show Fine-Grained Auditing results so far
--  the view is based on SYS.FGA_LOG$
-----
TTITLE 'Current Fine-Grained Auditing (FGA) Results'
COL audit_date      FORMAT A10      HEADING 'Audit|Date'
COL policy_name     FORMAT A16      HEADING 'Policy Name' WRAP
COL object_schema       FORMAT A10      HEADING 'Object|Schema'
COL object_name     FORMAT A20      HEADING 'Object Name' WRAP
COL db_user             FORMAT A10      HEADING 'DBUser'
COL sql_text        FORMAT A36      HEADING 'SQL Text' WRAP
SELECT
     TO_CHAR(timestamp,'dd/mm/yyyy hh24:mi:ss') audit_date
    ,db_user
    ,object_schema
    ,object_name
```

```
    ,policy_name
    ,sql_text
  FROM dba_fga_audit_trail
 ORDER BY timestamp;


-----
-- Use the new combined audit trail view (DBA_COMMON_AUDIT_TRAIL)
-- to see results of both Standard (i.e. AUDIT) and Fine-Grained
-- Auditing (i.e. via DBMS_FGA)
-----
COL audtype         FORMAT A03      HEADING 'Aud|Typ'
COL db_user             FORMAT A10      HEADING 'DBUser'
COL object_schema       FORMAT A06      HEADING 'Object|Schema'
COL object_name     FORMAT A20      HEADING 'Object Name' WRAP
COL policy_name     FORMAT A16      HEADING 'Policy Name' WRAP
COL audit_date      FORMAT A10      HEADING 'Audit|Date'
COL sql_text        FORMAT A32      HEADING 'SQL Text' WRAP
SELECT
     DECODE(audit_type,
            'Fine Grained Audit', 'FGA'
           ,'Standard Audit', 'STD'
           ,'UNK') audtype
    ,db_user
    ,object_schema
    ,object_name
    ,policy_name
    ,TO_CHAR(extended_timestamp,'mm/dd/yyyy hh24:mi:ss') audit_date
    ,sql_text
  FROM dba_common_audit_trail
 WHERE db_user NOT IN ('SYS','SYSTEM','DBSNMP','SYSMAN')
 ORDER BY extended_timestamp, db_user, object_schema, object_name;

-----
BEGIN
    DBMS_FGA.ADD_POLICY(
         object_schema => 'AP' -- if null, logon user schema
        ,object_name => 'VENDORS'
        ,policy_name => 'VENDORS_LO'
        ,audit_condition => 'ACTIVE_IND <> ''Y''' -- if NULL=TRUE
        ,audit_column => 'ACTIVE_IND,CREDIT_CARD,CREDIT_LIMIT'
        ,handler_schema => NULL
        ,handler_module => NULL -- the procedure will fire on audit
        ,enable => TRUE -- default is TRUE
        ,statement_types => 'SELECT'
        ,audit_trail => DBMS_FGA.DB_EXTENDED
        ,audit_column_opts => DBMS_FGA.ANY_COLUMNS -- or DBMS_FGA.ALL_COLUMNS
    );
END;
/

BEGIN
    DBMS_FGA.ADD_POLICY(
         object_schema => 'AP'
        ,object_name => 'RV_INVOICE_DETAILS'
        ,policy_name => 'RV_INVOICE_LO'
        ,audit_condition => NULL
        ,audit_column =>
'VENDOR_NAME,INVOICE_ID,EXTENDED_AMT,VENDOR_CREDIT_LIMIT'
        ,handler_schema => NULL
        ,handler_module => NULL
        ,enable => FALSE
        ,statement_types => 'SELECT'
```

```
        ,audit_trail => DBMS_FGA.DB_EXTENDED
        ,audit_column_opts => DBMS_FGA.ALL_COLUMNS
    );
END;
/



/*
|| Listing 1.5: FGA Policy Maintenance
*/

BEGIN
    -----
    -- Disabling an enabled, existing FGA policY
    -----
    DBMS_FGA.DISABLE_POLICY(
         object_schema => 'AP'
        ,object_name => 'INVOICES'
        ,policy_name => 'INVOICES_HI'
    );

    -----
    -- Dropping an enabled, existing FGA policY
    -----
    DBMS_FGA.DROP_POLICY(
        ,policy_name => 'RV_INVOICE_LOW'
        ,object_name => ' RV_INVOICE_DETAILS'
        ,policy_name => 'RV_INVOICE_LO'
    );
END;
/
```

Using Data Encryption

Oracle Transparent Data Encryption (TDE)

- **Caution**: Wallet file must be included in your backup.
- Column length changes on disk. Actual lengths not reported by DUMP or VSIZE.
- The Wallet must be opened after instance restart.

```
/* Setting up TDE */
-- 1.   Create the Wallet file:
-- add the following to the sqlnet.ora
ENCRYPTION_WALLET_LOCATION =
 (SOURCE=
   (METHOD=file)
     (METHOD_DATA=
(DIRECTORY=C:\oracle\OraDb10g\admin\ora10g\wallet)))

-- 2.   Set the master key: this is done only once:
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY <password>;

-- 3.   Create tables that contain encrypted columns
-- possible algorithms are AES128, (AES192), AES256, or 3DES168
-- the salt increases the protection but prevents indexing on the column.
CREATE TABLE emp (
 first_name VARCHAR2(128),
 ...
 empID NUMBER ENCRYPT NO SALT,
 salary NUMBER(6) ENCRYPT USING '3DES168',
 comm NUMBER(6) ENCRYPT);
ALTER TABLE EMP MODIFY ( SAL ENCRYPT NO SALT );
DESC EMP

/* Existing Tables and TDE */
--Add encrypted columns:
ALTER TABLE emp ADD (ssn VARCHAR2(11) ENCRYPT);
Encrypt unencrypted columns:
ALTER TABLE emp MODIFY (first_name ENCRYPT);
Disable column encryption:
ALTER TABLE emp MODIFY (first_name DECRYPT);
--Add or remove salt:
ALTER TABLE emp MODIFY (first_name ENCRYPT [NO] SALT);
--Change keys and the encryption algorithm:
ALTER TABLE emp REKEY USING '3DES168';

-- To Test TDE
SELECT
DBMS_ROWID.ROWID_TO_ABSOLUTE_FNO (ROWID,USER,'EMP'),
DBMS_ROWID.ROWID_BLOCK_NUMBER (ROWID)
```

```
FROM EMP;
File Nu Block Number
------- ------------
4       63
ALTER SESSION SET EVENTS  '10389 trace name context forever, level 1';
ALTER SYSTEM DUMP DATAFILE 4 BLOCK 63;

/* Opening and Closing the Wallet */
-- The Wallet must be opened after instance restart.
ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY password>
ALTER SYSTEM SET ENCRYPTION WALLET CLOSE
-- To verify that a wallet is open, you can query the V$ENCRYPTION_WALLET:
select WRL_PARAMETER, STATUS
from V$ENCRYPTION_WALLET
```

Tablespace Encryption

In Oracle Database 11g, you can encrypt an entire tablespace.

Encrypted Tablespace Limitations

- You cannot encrypt an existing tablespace.
- `exp` and `imp` utilities are not supported with objects in the encrypted tablespaces. Whereas `expdp` and `impdp` utilities are supported.
- You cannot re-create the tablespace encryption key.
- The `NO SALT` option is not supported.
- Temporary and undo tablespaces cannot be encrypted.
- You cannot transport an encrypted tablespace to a database that already has an Oracle wallet for TDE. In this case, use Oracle Data Pump to export the objects in the tablespace using the `expdp` with `ENCRYPTION_MODE=password` and then import them to the destination database.
- BFILES and external tables are not encrypted.
- Logically, encrypted tablespace is less efficient than normal un-encrypted tablespace.

Caution Losing the master key or the wallet file will lead to losing the data in the encrypted tablespace.

Encrypting a Tablespace

Create and open a wallet file, as explained in the previous section.

The tablespace creation statement for an encrypted tablespace has the following syntax:

```
CREATE TABLESPACE <tbsp_name> ...
[ENCRYPTION [USING <ALGORITHM>]] -- specify encryption algorithm
DEFAULT STORAGE(ENCRYPT)  -- encrypt objects in the tablespace

CREATE TABLESPACE tbsp1
DATAFILE '/u01/app/oracle/test/tbsp1_01.dbf' SIZE 500m
ENCRYPTION
```

```
DEFAULT STORAGE (ENCRYPT);

CREATE TABLESPACE mytbsp2
DATAFILE '/u01/app/oracle/test/mytbsp2_01.dbf' size 500m
ENCRYPTION USING '3DES168'
DEFAULT STORAGE (ENCRYPT);
```

The ALGORITHM clause accepts one of the following values:

- AES192 Advanced Encryption Standard (the default).
- 3DES168 Triple Data Encryption Standard 168-bit encryption
- AES128 Advanced Encryption Standard 128-bit encryption
- AES256 Advanced Encryption Standard 256-bit encryption

To know whether an existing tablespace is encrypted or not, issue the following query:

```
select vt.NAME, vet.ENCRYPTIONALG, vet.ENCRYPTEDTS
from   V$ENCRYPTED_TABLESPACES vet, V$TABLESPACE vt
where  vet.TS#=vt.TS#

SELECT tablespace_name, encrypted
2 FROM dba_tablespaces;
```

Fine-Grained Access Control for UTL_* Packages

Oracle Database 11g provides a mechanism to refine the level of access to the network access packages UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, and UTL_INADDR.

Creating ACL

You can use the DBMS_NETWORK_ACL_ADMIN package to facilitate management of the UTL_* network access packages as in the following steps:

1) Create an Access Control List (ACL): All ACL definitions are stored in XML DB in the form of XML documents. The ACL XML files reside in the /sys/acls directory of the XML DB repository. Following is an example of using the CREATE_ACL procedure to create an XML file called dba.xml:

```
begin
  DBMS_NETWORK_ACL_ADMIN.CREATE_ACL (
   ACL => 'dba.xml', -- case sensitive
   DESCRIPTION=> 'Network Access Control for the DBAs',
   PRINCIPAL  => 'SCOTT', -- user or role the privilege is granted or denied
(upper case)
   IS_GRANT   => TRUE, -- privilege is granted or denied
   PRIVILEGE  => 'connect', -- or 'resolve' (case sensitive)
   START_DATE => null,  -- when the access control entity ACE will be valid
   END_DATE   => null); -- ACE expiration date (TIMESTAMP WITH TIMEZONE format)
end;
```

Regarding the PRIVILEGE parameter, the database user needs the connect privilege to an external network host computer if he or she is connecting using the UTL_TCP, UTL_HTTP, UTL_SMTP, and UTL_MAIL utility packages. To resolve a host name that was given as a host IP address, or the IP address that was given as a host name, with the UTL_INADDR package, grant the database user the resolve privilege.

You can then query the RESOURCE_VIEW view to find the dba.xml ACL in the /sys/acls directory:

```
select ANY_PATH
from   RESOURCE_VIEW
where  ANY_PATH LIKE '/sys/acls/dba%'
```

Too may entries in the ACL may lead to significant XML DB performance drop because ACL are checked for each access to Oracle XML DB repository. As general rule of thumb, ACL check operations perform best when the number of ACEs in the ACL is at 16 entries or less.

2) Add Access Control Entries: Once you create the initial ACL, you can continue to add more privileges to the XML file. The following example will add the user RAMI to the dba.xml file and grant him network access:

```
begin
 DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE (
 ACL => 'dba.xml',
 PRINCIPAL => 'RAMI',
 IS_GRANT => TRUE,
```

```
 PRIVILEGE => 'connect',
 START_DATE => null, -- if the time interval is defined,
 END_DATE => null);  --  the ACE will expire after the specified date range
end;
/
COMMIT;
```

In ACL, the security entries are evaluating in order precedence. If you have two contradicting entries in the list, the first one in the order will take effect. You can control the order number of an added entry as follows:

```
begin
 DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE (
  POSITION => 1, -- on the top
  ACL => 'dba.xml',    PRINCIPAL => 'SAMI',
  IS_GRANT => FALSE,   PRIVILEGE => 'connect',
  START_DATE => null,  END_DATE => null);
end;
```

3) Assign Hosts: The ASSIGN_ACL procedure is used to authorize access to one or more network hosts as follows:

```
begin
DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL (
 ACL => 'dba.xml',   HOST => 'dbaexpert.com',
 LOWER_PORT => 80,   UPPER_PORT => 443);
end;
COMMIT;
```

The lower port and the upper port define the lower and the upper boundaries of the allowable port range. They should be set for connect privileges not resolve privileges.

4) Validate that the ACL permissions worked accordingly. Following is an example to test the code in the previous step.

```
select UTL_HTTP.REQUEST('http://www.ahmedbaraka.com') from dual;
```

If the sufficient ACL privileges or ACL assignments are not provided, you will receive the ORA-24247 error.

Access Control Lists Maintenance

Use DELETE_PRIVILEGE to remove an access control entry from the XML file.

```
exec DBMS_NETWORK_ACL_ADMIN.DELETE_PRIVILEGE( ACL=>'dba.xml', PRINCIPAL=> 'RAMI');
```

Use the DROP_ACL procedure to remove the XML file from the /sys/acls directory as follows:

```
exec DBMS_NETWORK_ACL_ADMIN.DROP_ACL ( ACL=>'dba.xml' );
```

Query Your Access Control List

To display list of the ACLs created in the database, use the following query:

```
select HOST, LOWER_PORT, UPPER_PORT, ACL from DBA_NETWORK_ACLS
```

You can query the DBA_NETWORK_ACL_PRIVILEGES view to query network privileges granted or denied for the access control list as follows:

```
select PRINCIPAL, PRIVILEGE, IS_GRANT
from   DBA_NETWORK_ACL_PRIVILEGES
where  ACL like '%dba.xml'
```

Logged on users can use the following query to see their access entries in the dba.xml file:

```
select HOST, LOWER_PORT, UPPER_PORT, STATUS privilege
from   USER_NETWORK_ACL_PRIVILEGES
where HOST in
(select * from
    table(DBMS_NETWORK_ACL_UTILITY.DOMAINS('dbaexpert.com')))
and PRIVILEGE = 'connect'
order by DBMS_NETWORK_ACL_UTILITY.DOMAIN_LEVEL(host) desc, LOWER_PORT;
```

Part 5 Oracle Database Performance Tuning

Managing Performance Statistics

Managing OS Statistics

- CPU Statistics:
  - doesn't exceed 95% in total.
  - check DB share on CPU
- Virtual Memory Statistics:
  - validate that memory usage does not increase after the system has reached a steady state after startup.
- Disk I/O Statistics:
  - current response time should be between 5 to 20 ms for a single block IO
  - the length of the disk queues shouldn't exceed two.
- Network Statistics:
  - Look at the network round-trip ping time and the number of collisions. Investigate it, if the network is causing large delays in response time.

- Reducing Disk Contention:
  - Increase the number of disks in the storage system
  - Separate the database and the redo log files
  - For a large table, use partitions to reduce I/O
  - Stripe the data either manually or by using a RAID disk-striping system
  - Invest in cutting-edge technology, such as file caching, to avoid I/O bottlenecks
  - Consider using ASM

```
/* from the database */
-- to gather system stats
see Gathering System Statistics

-- all stats
select STAT_NAME, VALUE, OSSTAT_ID, COMMENTS, CUMULATIVE
from V$OSSTAT;

/* CPU */
-- one-hour history of the Host CPU Utilization
select BEGIN_TIME, END_TIME, GROUP_ID, METRIC_ID, METRIC_NAME, VALUE, METRIC_UNIT
from V$SYSMETRIC_HISTORY
where METRIC_NAME LIKE '%Host CPU%'

-- top session CPU-comsumers
SELECT
 n.username,
 s.sid,
 s.value
FROM v$sesstat s,v$statname t, v$session n
WHERE s.statistic# = t.statistic#
  AND n.sid = s.sid
  AND t.name='CPU used by this session'
```

```
  AND s.value <> 0
ORDER BY s.value desc;

-- Decomposition of Total CPU Usage
--  if the parsing or recursive CPU usage PCT is high, then tuning is required
--  Recursive CPU Usage is for data dict lookups and executing PL/SQL programs
SELECT name, value,
round(value/(select sum(value) from v$sysstat WHERE NAME IN ('CPU used by this
session', 'recursive cpu usage','parse time cpu'))*100,2) PCT
FROM V$SYSSTAT
 WHERE NAME IN ('CPU used by this session', 'recursive cpu usage','parse time
cpu')
order by value DESC


/* io tuning */
-- IO related waits
select
 EVENT,
 TOTAL_WAITS_FG,
 TOTAL_TIMEOUTS_FG,
 TIME_WAITED_FG,
 AVERAGE_WAIT_FG,
 WAIT_CLASS,
 TOTAL_WAITS,
 TOTAL_TIMEOUTS,
 TIME_WAITED,
 AVERAGE_WAIT
from
V$SYSTEM_EVENT
WHERE WAIT_CLASS = 'User I/O'
order by  WAIT_CLASS;

-- io stats
select
 FILE_NO,
 FILETYPE_NAME,
 SMALL_READ_MEGABYTES "Single-block MegaBytes Reads",
 SMALL_WRITE_MEGABYTES "Single-block MegaBytes Writes",
 SMALL_READ_REQS "Single-block Read Requests",
 SMALL_WRITE_REQS "Single-block Write Requests",
 SMALL_READ_SERVICETIME "Total S-Block Read Time",
 SMALL_WRITE_SERVICETIME "Total S-Block Write Time",
 -- decode(SMALL_READ_REQS,0,0,SMALL_READ_SERVICETIME/SMALL_READ_REQS) "Per S-
Block Read Response T",
 SMALL_SYNC_READ_REQS,
 SMALL_SYNC_READ_LATENCY "S-Block Sync Read Latency (ms)",
 LARGE_READ_MEGABYTES "Multi-block MegaBytes Reads",
 LARGE_WRITE_MEGABYTES "Multi-block MegaBytes Writes",
 LARGE_READ_REQS "Multi-block Read Requests",
 LARGE_WRITE_REQS "Multi-block Write Requests",
 LARGE_READ_SERVICETIME "Total M-Block Read Time",
 LARGE_WRITE_SERVICETIME "Total M-Block Write Time",
 ASYNCH_IO,
 RETRIES_ON_ERROR
from V$IOSTAT_FILE
order by FILE_NO
/
```

```
-- Datafiles IO since instance startup
SELECT d.name,
 f.phyrds reads,
 f.phywrts wrts,
 (f.readtim / decode(f.phyrds,0,-1,f.phyrds))/10 ReadRespoinseTime_ms,
 (f.writetim / decode(f.phywrts,0,-1,phywrts))/10 WriteRespoinseTime_ms,
 SINGLEBLKRDTIM/10 Single_Block_ReadTime_ms
 FROM
 v$datafile d, v$filestat f
 WHERE d.file# = f.file#
 ORDER BY d.name;

-- Datafiles IO History
SELECT
 f.snap_id,
 f.filename,
 f.phyrds reads,
 f.phywrts wrts,
 (f.readtim / decode(f.phyrds,0,-1,f.phyrds))/10 ReadRespoinseTime_ms,
 (f.writetim / decode(f.phywrts,0,-1,phywrts))/10 WriteRespoinseTime_ms,
 SINGLEBLKRDTIM/10 Single_Block_ReadTime_ms,
 wait_count, time waittime
 FROM DBA_HIST_FILESTATXS f
 ORDER BY f.snap_id desc, filename



/*  on Windows */
--Performance Monitor tool.
For Windows Server 2003, can be downloaded from:
http://www.microsoft.com/downloads/details.aspx?familyid=09115420-8c9d-46b9-a9a5-
9bffcd237da2&displaylang=en

-- see using Using Performance Tool in Windows
-- see using OS Watcher
-- see also Optimizing Windows Server


/* on Unix */
/* CPU */
# process per second
sar -c 2 10 -- for all CPUs without interval, since last reboot
sar -c –P 0 -- for first CPU

# CPU% utilization
sar -u 4 5

# from vmstat output
# us: user CPU Time %
sy: system CPU Time %
id: idle CPU Time %
wa: Waiting for IO CPU Time %

# alternatively
vmstat -s | grep "cpu ticks"
# CPU% utilization
```

```
iostat –c
iostat 4 5 –c -- 5 times every 4 seconds



/* Virtual Memory Statistics */
-- Obtain info about memory
cat /proc/meminfo

-- Page ins and page outs
vmstat -s | grep "pages paged"
sar -B 2 50 -- every 2 seconds for 50 times

-- Swap ins and swap outs
# from vmstat output: swpd (so (swapped out) or si) should ideally be 0
# alternatively:
vmstat –s -S M | grep "swap"

sar –r
sar -W

-- Active and inactive pages: you shouldn't have too few inactive memory pages:
vmstat -S M
# alternatively:
vmstat -s -S M | grep "memory"

sar –r
sar –R



/* Disk I/O Statistics */
# disk usage
df –h

# from vmstat output: bi (blocks in) bo (blocks out)
# alternatively:
#vmstat -s -S M | grep "pages"
# reports disk statistics
vmstat -d | grep d
# tps: transfer per second
iostat –d
iostat 4 5 –d -- 5 times every 4 seconds
iostat -d –k  -- display output in kilobytes instead of blocks
iostat -d –x  -- extended report

# rtps wtps read/write requests per second issued to the physical disk
# bread/s bwrtn/s data read/write  from the drive in blocks per second
sar –b 5 10
# bloc device usage
# number of sectors (512 byte) read/written per second
#  if avque is greater larger than 1, disk contention is there
sar -d



/* Network Statistics */
```

```
# ping and check the latency
ping ...

# sar -n DEV | EDEV | SOCK | FULL
# DEV= network devices
rxpck/s packets received per second
txpck/s packets transmitted per second
rxbyt/s bytes received per second
txbyt/s bytes transmitted per second
# EDEV = failures from the network devices
rxerr/s bad packets received per second.
txerr/s errors that happened per second while transmitting packets
coll/s  collisions that happened per second while transmitting packets
rxdrop/s received packets dropped per second because of a lack of space in linux
buffers
txdrop/s transmitted packets dropped per second because of a lack of space in
linux buffers
txcarr/s carrier-errors that happened per second while transmitting packets
rxfram/s frame alignment errors that happened per second on received packets.
rxfifo/s FIFO overrun errors that happened per second on received packets.
txfifo/s FIFO overrun errors that happened per second on transmitted packets.
# SOCK =  sockets
totsck Total number of used sockets.
tcpsck Number of TCP sockets currently in use.
udpsck Number of UDP sockets currently in use.
rawsck Number of RAW sockets currently in use.
ip-frag Number of IP fragments currently in use.

#  display network interfaces
netstat –i
# summary stats on each protocol
netstat –s | less
```

Managing Database Statistics

- System and Session Statistics (and their executed statements)
- Time Model Statistics
- Wait Events
- Active Session History

System and Session Statistics

```
/* System stats */
select
 NAME,
 decode(CLASS,
'1','User','2','Redo','4','Enqueue','8','Cache','16','OS','32','RAC','64','SQL','1
28','Debug', CLASS) STAT_CLASS,
 VALUE
from V$SYSSTAT
order by NAME;


/* Session stats */
select
```

```
 T.SID,S.USERNAME, S.MACHINE, S.MODULE, S.ACTION,
 N.NAME,
 decode(N.CLASS,
'1','User','2','Redo','4','Enqueue','8','Cache','16','OS','32','RAC','64','SQL','1
28','Debug', N.CLASS) STAT_CLASS,
 VALUE
from V$SESSTAT T, V$STATNAME N, V$SESSION S
WHERE T.STATISTIC#=N.STATISTIC# and T.SID=S.SID
  and S.USERNAME NOT IN ('SYSTEM','SYS','DBSNMP','SYSMAN')
order by S.SID , N.NAME;


/* SQL Stats */
high buffer gets = using the wrong index, the wrong driving table in a join, or a
similar SQL-related error
buffer gets and disk reads are at identical levels = a missing index


-- top io consumers
SELECT executions, buffer_gets, disk_reads, rows_processed,SORTS,  sql_text
 FROM V$SQL
 WHERE buffer_gets > 100000 OR disk_reads > 100000
 ORDER BY buffer_gets + 100*disk_reads DESC;


-- top CPU consumers
SELECT executions,
  ROUND(elapsed_time/1000000, 2) elapsed_seconds,
  ROUND(cpu_time/1000000, 2) cpu_secs ,
  sql_text
  from (select * from v$sql order by elapsed_time desc)
 WHERE rownum <6
```

Time Model Statistics

- DB Time is an indicator instance workload.
- DB Time = CPU time + non-idle Wait time (of all the sessions accumulatively)

```
-- how long since the instance started
select STARTUP_TIME , ROUND((SYSDATE-STARTUP_TIME)*24,2) HOURS
from v$instance;

-- system-wide time-based stat
SELECT STAT_NAME "Stat. Name",
 round(VALUE/1000000) "Value (s)" ,
 round(VALUE/1000000/60) "Value (min)"
FROM V$SYS_TIME_MODEL;


-- SESSION-wide time-based stats
SELECT
  E.SID, S.USERNAME, S.MACHINE, S.MODULE, S.ACTION,
  STAT_NAME "Stat. Name",
  round(VALUE/1000000) "Value (s)" ,
  round(VALUE/1000000/60) "Value (min)"
FROM  V$SESS_TIME_MODEL E, V$SESSION S
WHERE E.SID = S.SID
  and S.USERNAME NOT IN ('SYSTEM','SYS','DBSNMP','SYSMAN')
```

```
order by E.SID;
```

Wait Events

- The wait events are only the symptoms of problems, most likely within the application code.
- After defining the troubled waiting event, you can get further info by tracing the suspected session.

```
-- TIMED_STATISTICS must be true (default)
show parameter TIMED_STATISTICS

/* Wait time for the whole instance */
-- System metrics captured in recent 60-sec or 15-sec
--  If 'Database Wait Time Ratio' is higher than 'Database CPU Time Ratio',
--   consider looking for bottlenecks
select
 GROUP_ID,
 METRIC_NAME,
 VALUE,
 METRIC_UNIT,
 INTSIZE_CSEC/100 Interval_Duration,
 TO_CHAR(BEGIN_TIME,'HH24:MI:SS') BEGIN_TIME,
 TO_CHAR(END_TIME,'HH24:MI:SS') END_TIME
from V$SYSMETRIC
-- where METRIC_NAME IN ('Database Wait Time Ratio','Database CPU Time Ratio')
order BY END_TIME DESC

-- in the recent hour
SELECT
 GROUP_ID,
 METRIC_NAME,
 VALUE,
 METRIC_UNIT,
 ROUND(INTSIZE_CSEC/100) Interval_Duration,
 TO_CHAR(BEGIN_TIME,'HH24:MI:SS') BEGIN_TIME,
 TO_CHAR(END_TIME,'HH24:MI:SS') END_TIME
FROM V$SYSMETRIC_HISTORY
 WHERE METRIC_NAME IN ('Database Wait Time Ratio','Database CPU Time Ratio')
ORDER BY END_TIME DESC


/* WAIT EVENTS */
-- waite events in the instance:
--  Top Wait Classes By Instance Total
--  AWR reports could also assist you
select
WAIT_CLASS,
TIME_WAITED, round(TIME_WAITED/TOT_WAIT*100,2) TIME_WAITED_PCT, TIME_WAITED_FG,
 round(TIME_WAITED_FG/TOT_WAIT*100,2) TIME_WAITED_FG_PCT
from V$SYSTEM_WAIT_CLASS, (select sum(TIME_WAITED) TOT_WAIT from
V$SYSTEM_WAIT_CLASS where WAIT_CLASS <>'Idle'),
 (select sum(TIME_WAITED_FG) TOT_WAIT_FG from V$SYSTEM_WAIT_CLASS where WAIT_CLASS
<>'Idle')
where WAIT_CLASS <>'Idle'
order by  TIME_WAITED_FG_PCT DESC
```

```
-- Wait Classes by Instance Wide in the Last Hour
select
 TO_CHAR(BEGIN_TIME,'HH24:MI') BEGIN_TIME,
 TO_CHAR(END_TIME,'HH24:MI') END_TIME,
 INTSIZE_CSEC/100 Interval_sec,
 WAIT_CLASS#,
 (SELECT DISTINCT WAIT_CLASS FROM V$EVENT_NAME X WHERE
X.WAIT_CLASS#=W.WAIT_CLASS#) WAIT_CLASS_NAME,
 TIME_WAITED, DBTIME_IN_WAIT "Pct of DB Time spent", WAIT_COUNT
from V$WAITCLASSMETRIC_HISTORY W
where WAIT_CLASS#<>6 -- Idle
order by BEGIN_TIME DESC, TIME_WAITED DESC;


-- Wait Events by Instance Total
--  typically waits by foreground processes are what we care about
select
 EVENT,
 TIME_WAITED_FG,
 ROUND(TIME_WAITED_FG/TOT_WAIT_FG*100,2) TIME_WAITED_PCT,
 TOTAL_WAITS_FG,
 TOTAL_TIMEOUTS_FG,
 AVERAGE_WAIT_FG,
 WAIT_CLASS,
 TOTAL_WAITS,
 TOTAL_TIMEOUTS,
 TIME_WAITED,
 AVERAGE_WAIT
from V$SYSTEM_EVENT, (SELECT SUM(TIME_WAITED_FG) TOT_WAIT_FG FROM V$SYSTEM_EVENT
where WAIT_CLASS <> 'Idle')
where WAIT_CLASS <> 'Idle'
order by TIME_WAITED_FG DESC;

-- if buffer busy was on the top waits, get more info about
-- block contention statistics
select CLASS, TIME, COUNT
from V$WAITSTAT
order by TIME desc


-- Wait Events by Sessions
--  detailed information on V$SESSION ( or V$SESSION_WAIT )
--  cols in V$SESSION_WAIT already contained in V$SESSION
select
 E.SID, S.USERNAME, S.MACHINE, S.MODULE, S.ACTION,
 S.STATUS,
 E.WAIT_CLASS,
 E.EVENT,
 S.STATE,
 WAIT_TIME,
 TIME_WAITED,
 SECONDS_IN_WAIT,
 AVERAGE_WAIT,
 P1TEXT,
 P1,
 P2TEXT,
 P2,
```

```
 P3TEXT,
 P3,
 TOTAL_WAITS,
 MAX_WAIT,
 TOTAL_TIMEOUTS,
 ROW_WAIT_OBJ# WAITED_OBJECT,
 TO_CHAR(S.LOGON_TIME,'HH24:MI') LOGON_TIME,
 S.BLOCKING_SESSION
from V$SESSION_EVENT E, V$SESSION S
where E.SID = S.SID
  and S.USERNAME NOT IN ('SYSTEM','SYS','DBSNMP','SYSMAN')
  and E.WAIT_CLASS <> 'Idle'
order by E.TIME_WAITED desc
/


-- if the problem in the latch, further details can be obtained:
SELECT
 ADDR Latch_Object_Address,
 LATCH#,
 LEVEL#,
 NAME,
 HASH,
 GETS,
 MISSES,
 case misses when 0 then 0 else misses/(misses+GETS+SLEEPS) end MISSES_RATIO,
 SLEEPS,
 IMMEDIATE_GETS,
 IMMEDIATE_MISSES,
 SPIN_GETS,
 WAIT_TIME WAIT_TIME_US
FROM V$LATCH
order by MISSES desc
```

Active Session History (ASH)

- Used when you want to analyze for a previous period not taken yet by AWR (let's say last 10 minutes).
- V$ACTIVE_SESSION_HISTORY is flushed into DBA_HIST_ACTIVE_SESS_HISTORY when AWR is taken (every hour by default).

```
-- ASH list
SELECT SAMPLE_ID, SAMPLE_TIME, SESSION_ID, SESSION_SERIAL#, SESSION_TYPE,
 FLAGS, USER_ID, SQL_ID, SQL_CHILD_NUMBER, SQL_OPCODE, FORCE_MATCHING_SIGNATURE,
 TOP_LEVEL_SQL_ID, TOP_LEVEL_SQL_OPCODE, SQL_PLAN_HASH_VALUE, SQL_PLAN_LINE_ID,
SQL_PLAN_OPERATION, SQL_PLAN_OPTIONS, SQL_EXEC_ID, SQL_EXEC_START,
PLSQL_ENTRY_OBJECT_ID, PLSQL_ENTRY_SUBPROGRAM_ID, PLSQL_OBJECT_ID,
PLSQL_SUBPROGRAM_ID, QC_INSTANCE_ID, QC_SESSION_ID, QC_SESSION_SERIAL#,
 EVENT, EVENT_ID, EVENT#, SEQ#, P1TEXT, P1, P2TEXT, P2, P3TEXT, P3, WAIT_CLASS,
WAIT_CLASS_ID, WAIT_TIME,  SESSION_STATE, TIME_WAITED, BLOCKING_SESSION_STATUS,
BLOCKING_SESSION, BLOCKING_SESSION_SERIAL#,  CURRENT_OBJ#, CURRENT_FILE#,
CURRENT_BLOCK#, CURRENT_ROW#, CONSUMER_GROUP_ID, XID,  REMOTE_INSTANCE#,
IN_CONNECTION_MGMT, IN_PARSE, IN_HARD_PARSE, IN_SQL_EXECUTION,
IN_PLSQL_EXECUTION, IN_PLSQL_RPC, IN_PLSQL_COMPILATION, IN_JAVA_EXECUTION,
IN_BIND,  IN_CURSOR_CLOSE, SERVICE_HASH, PROGRAM, MODULE, ACTION, CLIENT_ID
FROM V$ACTIVE_SESSION_HISTORY
Where SESSION_TYPE <> 'BACKGROUND'
and USER_ID  not in ( select u.user_id from dba_users u where username IN
```

```
('SYS','SYSTEM','DBSNMP','SYSMAN') )
order by Sample_Time desc


-- ASH report
-- info about the SQL that ran during the time you specify
$ORACLE_HOME/rdbms/admin/ashrpt.sql
The ashrpti.sql lets you specify the db instance


-- Top Waited-for Objects
SELECT (SELECT o.object_name
        FROM   dba_objects o
        WHERE  o.object_id = current_obj#) object_name,
       (SELECT o.object_type
        FROM   dba_objects o
        WHERE  o.object_id = current_obj#) object_type,
       a.event,
       SUM (a.wait_time + a.time_waited)   total_wait_time
FROM   v$active_session_history a
WHERE  ( a.wait_time + a.time_waited ) > 0
       AND a.current_obj# IS NOT NULL
       AND a.sample_time BETWEEN SYSDATE - 15 / 1440 AND SYSDATE
GROUP  BY a.event,
          current_obj#
ORDER  BY total_wait_time DESC;


-- Top Waits
SELECT a.event,
 SUM(a.wait_time + a.time_waited) total_wait_time
FROM v$active_session_history a
WHERE a.sample_time between
 sysdate - 30/2880 and sysdate
GROUP BY a.event
ORDER BY total_wait_time DESC;


-- Top Waiting Users
SELECT s.sid, s.username,
 SUM(a.wait_time +
 a.time_waited) total_wait_time
FROM v$active_session_history a,
 v$session s
WHERE a.sample_time between sysdate - 30/2880 and sysdate
 AND a.session_id=s.sid
 GROUP BY s.sid, s.username
ORDER BY total_wait_time DESC;


-- Top Waiting SQL Statements
SELECT a.user_id,d.username,s.sql_text,
 SUM(a.wait_time + a.time_waited) total_wait_time
FROM v$active_session_history a,
 v$sqlarea s,
 dba_users d
 WHERE a.sample_time between sysdate - 30/2880 and sysdate
```

```
 AND a.sql_id = s.sql_id
 AND a.user_id = d.user_id
GROUP BY a.user_id,s.sql_text, d.username;

-- Top Resource Consuming SQL
SELECT hash_value, executions,
 ROUND (elapsed_time/1000000, 2) total_time,
 ROUND (cpu_time/1000000, 2) cpu_seconds
 FROM (SELECT * FROM V$SQL
 ORDER BY elapsed_time desc);

-- ASH history
SELECT *
FROM DBA_HIST_ACTIVE_SESS_HISTORY
Where SESSION_TYPE <> 2  -- 'BACKGROUND' (SESSION_TYPE in V$ is varchar2)
order by SNAP_ID desc
```

Segment Statistics

- You drill down into segment stats from the instance, sessions or ASM wait stats.

```
select OWNER, OBJECT_TYPE, OBJECT_NAME, TABLESPACE_NAME, STATISTIC_NAME, VALUE
from V$SEGMENT_STATISTICS
-- WHERE statistic_name='buffer busy waits'
order by value desc;
```

Handling Important Oracle Wait Events

Buffer Busy Waits	Check type of block is causing the wait. Use locally managed tablespaces with ASSM. Consider using global hash-partitioned indexes. Tune SQL statements as necessary to fix these waits. Check that the sar -d utility might indicate high request queues and service times. Check your core dump directory is not too big.
db file scattered read	Reduce the demand for physical I/Os: Raising the buffer cache component, add missing indexes on key tables, Optimize SQL statements. Increase the capacity of the system to handle more I/Os.

Db File Sequential Read	Indicate index usage.
	Increase PGA_AGGREGATE_TARGET.
	If the objects aren't too large, you can use the DEFAULT and KEEP buffer pools to retain them in memory.
	When you have file# and Block# from wait details, you can get the object name:
	`select segment_name from dba_extents`
	`where file_id = <file#> and <block#> between block_id and block_id + blocks - 1  and rownum = 1`

Direct Path Read and Direct Path Write	It occurs while performing a direct read or write into the PGA, bypassing the SGA buffer cache. Enable PGA auto tuning. Increase the number of disks.

Free Buffer Waits	The number of dirty buffers in cache too high for the db writer. Increase Db Buffer Cache Increase DB_WRITER_PROCESSES (one per CPU) Check the IO

Enqueue Waits	Sessions are waiting for locks held by other sessions.
	Query V$ENQUEUE_STAT
	`select * from V$ENQUEUE_STAT`
	`where CUM_WAIT_TIME<>0 OR FAILED_REQ#<>0`
	May caused by infrequent commits and dictionary managed tbs.

Latch Free	`SELECT a.name "Latch Name",`
	`a.gets "Gets (Wait)",`
	`a.misses "Misses (Wait)",`
	`(1 - (misses / gets)) * 100 "Latch Hit Ratio %"`
	`FROM V$LATCH a`
	`WHERE a.gets != 0`
	`UNION`
	`SELECT a.name "Latch Name",`
	`a.gets "Gets (Wait)",`
	`a.misses "Misses (Wait)",`
	`100 "Latch Hit Ratio"`
	`FROM V$LATCH a`
	`WHERE a.gets = 0`
	`ORDER BY 4;`
	`shared pool latch` (and the `library cache latches`) reasons:
	• An undersized shared pool
	• Failure to use bind variables
	• Using dissimilar SQL statements and failing to reuse statements
	• Users frequently logging off and logging back into the application
	• Failure to keep cursors open after each execution
	• Using a shared pool size that's too large
	`cache buffers LRU chain` and `cache buffer chain` may be caused by table full scan or unselective indexes.

Log Buffer Space	Check I/O to the redo log disk Consider increasing LOG_BUFFER (so that Log File Sync wait event doesn't increase as well) For a materialized view with COMPLETE refresh, consider setting ATOMIC_REFRESH=false in DBMS_MVIEW.REFRESH to avoid logging
Log File Switch	Check the archive destination isn't full. Increase sizes of redo log files. Increase the number of archiver (ARCn) processes. Small values lead to "redo log space requests" in V$SYSSTAT
Log File Sync	Too-frequent commits Redo Log I/O bottleneck
library cache pin	It occurs when you are compiling or parsing a PL/SQL object or a view and Oracle wants to pin an object in memory in the library cache and

	ensure no other processes can update the object at the same time. This is my script for tracking down who is blocking who in the event of a library cache pin event: `select decode(lob.kglobtyp, 0, 'NEXT OBJECT', 1, 'INDEX', 2, 'TABLE', 3, 'CLUSTER', 4, 'VIEW', 5, 'SYNONYM', 6, 'SEQUENCE', 7, 'PROCEDURE', 8, 'FUNCTION', 9, 'PACKAGE', 11, 'PACKAGE BODY', 12, 'TRIGGER', 13, 'TYPE', 14, 'TYPE BODY', 19, 'TABLE PARTITION', 20, 'INDEX PARTITION', 21, 'LOB', 22, 'LIBRARY', 23, 'DIRECTORY', 24, 'QUEUE', 28, 'JAVA SOURCE', 29, 'JAVA CLASS', 30, 'JAVA RESOURCE', 32, 'INDEXTYPE', 33, 'OPERATOR', 34, 'TABLE SUBPARTITION', 35, 'INDEX SUBPARTITION', 40, 'LOB PARTITION', 41, 'LOB SUBPARTITION', 42, 'MATERIALIZED VIEW', 43, 'DIMENSION', 44, 'CONTEXT', 46, 'RULE SET', 47, 'RESOURCE PLAN', 48, 'CONSUMER GROUP', 51, 'SUBSCRIPTION', 52, 'LOCATION', 55, 'XML SCHEMA', 56, 'JAVA DATA', 57, 'SECURITY PROFILE', 59, 'RULE', 62, 'EVALUATION CONTEXT', 'UNDEFINED') object_type, lob.KGLNAOBJ object_name, pn.KGLPNMOD lock_mode_held, pn.KGLPNREQ lock_mode_requested, ses.sid, ses.serial#, ses.username FROM x$kglpn pn, v$session ses, x$kglob lob, v$session_wait vsw WHERE pn.KGLPNUSE = ses.saddr and pn.KGLPNHDL = lob.KGLHDADR and lob.kglhdadr = vsw.p1raw and vsw.event = 'library cache pin' order by lock_mode_held desc`
direct path read temp	This happens in: • The sorts are too large to fit in memory and some of the sort data is written out directly to disk. • Parallel slaves are used for scanning data. This is the biggest wait for large data warehouse sites. Consider setting sort_area_size or pga_aggreagate_target
enq: JI - contention	The session is waiting for a materialized view to finish its undergoing refresh. For example, you are trying to refresh two materialized views reading from same master table(s) in the same time. Another example, you are trying to compile a materialized view when another session is refreshing it.

List of Idle Waits Events

On Oracle 10g, the idle waits events are:

```
AQ Proxy Cleanup Wait
ASM background timer
DIAG idle wait
EMON idle wait
```

```
KSV master wait
LNS ASYNC archive log
LNS ASYNC dest activation
LNS ASYNC end of log
LogMiner: client waiting for transaction
LogMiner: slave waiting for activate message
LogMiner: wakeup event for builder
LogMiner: wakeup event for preparer
LogMiner: wakeup event for reader
Null event
PX Deq Credit: need buffer
PX Deq Credit: send blkd
PX Deq: Execute Reply
PX Deq: Execution Msg
PX Deq: Par Recov Execute
PX Deq: Signal ACK
PX Deq: Table Q Normal
PX Deq: Table Q Sample
PX Deque wait
PX Idle Wait
Queue Monitor Shutdown Wait
Queue Monitor Slave Wait
Queue Monitor Wait
SQL*Net message from client
SQL*Net message to client
SQL*Net more data from client
STREAMS apply coord waiting for slave message
STREAMS apply slave idle wait
STREAMS apply slave waiting for coord message
STREAMS capture process filter callback wait for ruleset
STREAMS fetch slave waiting for txns
STREAMS waiting for subscribers to catch up
Streams AQ: RAC qmn coordinator idle wait
Streams AQ: deallocate messages from Streams Pool
Streams AQ: delete acknowledged messages
Streams AQ: qmn coordinator idle wait
Streams AQ: qmn slave idle wait
Streams AQ: waiting for messages in the queue
Streams AQ: waiting for time management or cleanup tasks
Streams fetch slave: waiting for txns
class slave wait
client message
dispatcher timer
gcs for action
gcs remote message
ges remote message
i/o slave wait
jobq slave wait
knlqdeq
lock manager wait for remote message
master wait
null event
parallel query dequeue
pipe get
pmon timer
queue messages
rdbms ipc message
```

```
rdbms ipc reply
reliable message
slave wait
smon timer
virtual circuit status
wait for activate message
wait for unread message on broadcast channel
wakeup event for builder
wakeup event for preparer
wakeup event for reader
wakeup time manager
```

Using Performance Monitor Tool in Windows

- Instant monitor: click on the '+' icon to start up new counters. Click on 'Process' under 'Performance Object'. Select counters, 'Private Bytes' (what Task Manager shows), Thread Count, 'Handle Count', and 'Virtual Bytes' in the 'Select counters from list' window. Also select the process, 'Oracle' in the 'Select instance from list' window. Click on the 'Add' button.
- Scheduled monitor: click Counter Logs node, right click on the right hand side pane, select 'New Log Settings', type the name, select the objects to monitor, set the schedule and the generated log file.

Using OS Watcher for Windows

- OS Watcher for Unix and Linux, see Using OS Watcher for Linux.
- OS Watcher for Windows (OSWFW) is several batch files that run the Windows utility logman and schtasks.
- If OSWFW is installed on RAC, when it is first run, it will read the Cluster Registry and copy itself to the other nodes in a directory that is the same name as the directory it was first unzipped in.
- Its reference is Note 433472.1

```
/* Removing OSWFW */
oswatcher remove


/* Initially configure OSWFW */
OSWatcher {ARG1} {ARG2} {ARG3}
    ARG1 = Snapshot interval in seconds
    ARG2 = Runtime Interval - hours OSWatcher will run
    ARG3 = RAC - detect Real Application Cluster

OSWatcher 30 48


/* OSWatcher stop */
OSWatcher stop {node name}


/* Getting the Status of OSWFW */
OSWatcher status
```

```
/* Querying details of a specific counter or task */
OSWatcher query {node name} {counter / OSWCleanup / OSWPrivNet}
OSWatcher query all
oswatcher query OSWThread


/* Extracting the Names of the Counters in a Capture File */
relog {trace_file_name} -q
# to sort the output:
relog {trace_file_name} -q | sort /+1
```

Optimizing Performance in Windows Server

Optimizing Performance in Windows Server 2003

```
/* disable un-required services */
Usually, the following services are not required in a database server and can be
disabled:
  Alerter
  ClipBook
  Computer Browser
  DHCP Client
  DHCP Server
  Fax Service
  File Replication
  INfrared Monitor
  Internet Connection Sharing
  Messenger
  NetMeeting Remote Desktop Sharing
  Network DDE
  Network DDE DSDM
  NWLink NetBIOS
  NWLink IPX/SPX
  Print Spooler
  TCP/IP NetBIOS Helper Service: unless member of a domain
  Telephony
  Telnet
  Uninterruptible Power Supply


/* Optimize Performance for Background Services */
Oracle database runs as a background service.
To increase performance for background services, follow these steps:
 1. Click Start, click Control Panel, and then click System.
 2. Click the Advanced tab, and then click Settings under Performance.
 3. Click the Advanced tab, click Background services, and then click OK


/* Optimize Data Throughput for Network Applications */
 1. In Windows Explorer, right-click My Network Places, and then click Properties.
 2. Right-click the Local Area Connection that you want to optimize, then click
Properties.
 3. In the This connection uses the following items list, click (but do not clear
its check box) File and Printer Sharing for Microsoft Networks, and then click
Properties.
 4. Click Maximum data throughput for network applications, click OK, then Close.
```

Tuning the Database Instance

Tuning the Shared Pool

```
/* Data Dictionary Hit Ratio */
-- around 90 (depending on the application, low value in some cases acceptable)
SELECT (sum(gets - getmisses - fixed)) / SUM(gets) "data dictionary hit ratio"
from v$rowcache;



/* Pinning Objects in the Shared Pool (LC)*/
SELECT type, COUNT(*) OBJECTS,
 SUM(DECODE(KEPT,'YES',1,0)) KEPT,
 SUM(loads) - count(*) reloads
 FROM V$DB_OBJECT_CACHE
 GROUP BY type
 ORDER BY objects DESC;

select *
FROM V$DB_OBJECT_CACHE
order by loads desc;

-- pin objects with high loads:
--  un-retained after db restart
--  you can make script to load then when db starts
EXECUTE SYS.DBMS_SHARED_POOL.KEEP(NEW_EMP.PKG, PACKAGE);
EXECUTE SYS.DBMS_SHARED_POOL.UNKEEP(NEW_EMP.PKG,PACKAGE);


/* Measuring Library Cache Efficiency */
-- lib cache hit ratio
SELECT SUM(pinhits)/sum(pins) Library_cache_hit_ratio
 FROM V$LIBRARYCACHE;


-- statement reloads
SELECT namespace, pins, pinhits, reloads
 FROM V$LIBRARYCACHE
 ORDER BY namespace;


-- lib cache memory usage
select LC_NAMESPACE, LC_INUSE_MEMORY_OBJECTS, LC_INUSE_MEMORY_SIZE,
LC_FREEABLE_MEMORY_OBJECTS, LC_FREEABLE_MEMORY_SIZE
from V$LIBRARY_CACHE_MEMORY
order by LC_INUSE_MEMORY_OBJECTS desc, LC_INUSE_MEMORY_SIZE desc,
LC_FREEABLE_MEMORY_OBJECTS desc, LC_FREEABLE_MEMORY_SIZE desc;


-- to estimate the optimal shared pool size
-- you can just set the MEMORY_TARGET
select SHARED_POOL_SIZE_FOR_ESTIMATE, SHARED_POOL_SIZE_FACTOR, ESTD_LC_SIZE,
  ESTD_LC_MEMORY_OBJECTS, ESTD_LC_TIME_SAVED, ESTD_LC_TIME_SAVED_FACTOR,
  ESTD_LC_LOAD_TIME, ESTD_LC_LOAD_TIME_FACTOR, ESTD_LC_MEMORY_OBJECT_HITS
from V$SHARED_POOL_ADVICE
```

```
order by SHARED_POOL_SIZE_FOR_ESTIMATE desc;


-- if literal values rather than bind values are used
--  by the applications (high hard parse):
CURSOR_SHARING=FORCE (recommended) or SIMILAR (not EXACT)


/* Setting CURSOR_SPACE_FOR_TIME */
If CURSOR_SPACE_FOR_TIME=TRUE, you ensure that the cursors for the application
cannot be deallocated while the application cursors are still open. It will then
eliminate the Oracle's overhead to check whether the cursor is flushed from the
library cache.
It will result in increase in the shared pool memory.


/* Setting SESSION_CACHED_CURSORS  */
It ensures that for any cursor for which more than three parse requests are made,
the parse requests are automatically cached in the session cursor cache. It avoids
high soft parse.
Good to use in Forms-based apps.
ALTER SESSION SET SESSION_CACHED_CURSORS = value;

If the value of session cursor cache hits is low compared to the total parse count
for a session, then the SESSION_CACHED_CURSORS parameter value should be increased
(also good to make it larger than OPEN_CURSORS):

 select NAME, VALUE
 from v$sysstat
 where name = 'session cursor cache hits'

 select SID, NAME, VALUE
 from v$sesstat, V$STATNAME
 where V$SESSTAT.STATISTIC# = V$STATNAME.STATISTIC#
 and  NAME LIKE 'parse count (total)'
 order by VALUE desc

alter system set SESSION_CACHED_CURSORS=300 scope=spfile ;
```

Tuning the Buffer Cache

```
/* Sizing the Buffer Cache */
-- use MEMORY_TARGET
-- size assigned to it
select current_size/1024/1024
from V$SGA_DYNAMIC_COMPONENTS
where component = 'DEFAULT buffer cache'


-- different areas in the buffer cache
select count(*) blocks, State
    from (
    select decode (state,
       0, 'Free',
       1, decode (lrba_seq,0, 'Available','Being Used'),
       3, 'Being Used',
```

```
          state) State
     from x$bh )
group by rollup(state)
order by count(*) desc;


-- to get the buffer cache hit ratio
SELECT NAME, PHYSICAL_READS, DB_BLOCK_GETS, CONSISTENT_GETS,
round( 1 - (PHYSICAL_READS/(DB_BLOCK_GETS + CONSISTENT_GETS)),4)*100 "HitRatio"
FROM V$BUFFER_POOL_STATISTICS;


/* Using Multiple Pools for the Buffer Cache */
-- x$bh: contains a record (the buffer header) for each block in the buffer
select count(*), State from (
    select decode (state,
       0, 'Free',
       1, decode (lrba_seq,0, 'Available','Being Used'),
       3, 'Being Used',
          state) State
     from x$bh )
group by state


/* Sizing the Buffer Cache */
-- use MEMORY_TARGET

-- to get the buffer cache hit ratio
SELECT NAME, PHYSICAL_READS, DB_BLOCK_GETS, CONSISTENT_GETS,
round( 1 - (PHYSICAL_READS/(DB_BLOCK_GETS + CONSISTENT_GETS)),4)*100 "HitRatio"
FROM V$BUFFER_POOL_STATISTICS;


/* Using Multiple Pools for the Buffer Cache */
-- To determine objects that are candidates for the recycle buffer pool
--  x$bh: contains a record (the buffer header) for each block in the buffer
select
 obj object,
 o.object_name,
 count(*) buffers,
 round((count(*)/totsize) * 100,2) percent_cache
FROM x$bh, (select count(*) totsize FROM x$bh ), DBA_OBJECTS o
WHERE ( tch=1 -- touch count
        OR (tch = 0 and lru_flag <10))
   and obj=o.object_id(+)
GROUP BY obj, o.object_name, totsize
having round((count(*)/totsize) * 100,2) > 1
order by percent_cache desc;

-- objects candidate for keep pool
--  objects of at least 25 buffers and have an average touch count of more than 5
SELECT obj object, o.object_name,
 count(*) buffers,
 AVG(tch) average_touch_count
FROM x$bh, dba_objects o
WHERE obj=o.object_id(+) and
 lru_flag = 8
```

```
GROUP BY obj, o.object_name
HAVING avg(tch) > 5 AND count(*) > 25
order by avg(tch) desc


-- define the areas
DB_KEEP_CACHE_SIZE=16MB
DB_RECYCLE_CACHE_SIZE=16MB


-- change the table's pool
ALTER TABLE test1 STORAGE (buffer_pool keep);
ALTER TABLE test2 STORAGE (buffer_pool recycle);
```

Tuning PGA

```
-- PGA stats
SELECT NAME, VALUE, UNIT FROM V$PGASTAT

-- PGA used by each session
SELECT
 s.sid,a.username, round(s.value/1024,2) KB
FROM
 V$SESSTAT S, V$STATNAME N, V$SESSION A
 WHERE
 n.STATISTIC# = s.STATISTIC# and
name = 'session pga memory'
AND s.sid=a.sid
 ORDER BY s.value desc;

-- PGA used by sessions
select PID, SERIAL#, CATEGORY, round(ALLOCATED/1024,2) ALLOCATED_KB1,
 USED, round(MAX_ALLOCATED/1024,2) MAX_ALLOCATED_KB
from V$PROCESS_MEMORY

-- PGA used by processes
SELECT
 program,
 pga_used_mem,
 pga_alloc_mem,
 pga_freeable_mem,
 pga_max_mem
from V$PROCESS;
```

Dumping the PGA

```
SQL>conn / as sysdba
SQL> alter system set max_dump_file_size=unlimited;
SQL> connect sys as sysdba
SQL> oradebug setospid <OSpid from query above for unix systems>

If on a MSwindows platform you will need to attach using the setorapid instead of
setospid
SQL> oradebug setorapid <Orapid from query above>
```

```
SQL> oradebug unlimit
SQL> oradebug dump errorstack 10
SQL> oradebug dump heapdump 536870917
SQL> oradebug tracefile_name (shows the path and filename information)
SQL> oradebug close_trace (closes the trace file)
SQL>exit
```

Using Server Result Cache

- For SQL Result Cache, if an underlying table is updated, the result cache will be invalidate and the statement will be re-executed.

Caution When a session reads from a PL/SQL function result cache, the function body is not executed. This means, if the function includes any IO or auditing code, this code will not actually be executed.

SQL Result Cache Restrictions:

- Queries against data dictionary objects and temporary tables are not supported.
- Queries that use the following SQL functions: CURRENT_DATE, CURRENT_TIMESTAMP, LOCAL_TIMESTAMP, USERENV/SYS_CONTEXT, SYS_GUID, SYSDATE and SYS_TIMESTAMP are not supported.
- Queries with bind variables can reuse a cached result only for identical variable values.
- Results of the queries retrieving non current version of data are not cached in the result cache.
- Results of the flashback queries are not cached.

Restrictions on PL/SQL Function Result Cache include:

- The function cannot be defined in a module using invoker's rights.
- The function cannot be used in an anonymous block.
- The function cannot have any OUT or IN OUT parameters.
- The function cannot have IN parameters that are BLOB, CLOB, NCLOB, REF CURSOR, collections, objects, or records.
- The function cannot return a BLOB, CLOB, NCLOB, REF CURSOR, OBJECTS, or records. It can return a collection as long as the collection does not contain one of these types.

```
/* Configuring Result Cache */
RESULT_CACHE_MAX_SIZE maximum amount of SGA memory (in bytes) that can be used by
the Result Cache (taken from shared pool).
If the value of this parameter is 0, then the feature is disabled.

show parameter RESULT_CACHE_MAX_SIZE

ALTER SYSTEM SET RESULT_CACHE_MAX_SIZE =8M;

The parameter RESULT_CACHE_MAX_RESULT specifies the percentage of
RESULT_CACHE_MAX_SIZE that any single result can use.
Its default value is five.
ALTER SYSTEM SET RESULT_CACHE_MAX_RESULT =25;
```

```
The parameter RESULT_CACHE_REMOTE_EXPIRATION specifies the number of minutes that
a result using a remote object is allowed to remain valid. Setting this parameter
to 0 (the default) implies that results using remote objects should not be cached.



/* Controlling Result Cache Behavior */
The RESULT_CACHE_MODE initialization parameter:
 MANUAL The ResultCache operator is added, only if you use the RESULT_CACHE hint in
the SQL query.
 FORCE  The ResultCache operator is added to the root of all SELECT statements, if
that is possible. However, if the statement contains a  NO_RESULT_CACHE hint, then
the hint takes precedence over the parameter setting.

ALTER SYSTEM SET RESULT_CACHE_MODE =FORCE;
ALTER SESSION SET RESULT_CACHE_MODE =FORCE;


SELECT /*+ result_cache */
 AVG(SALARY), E.DEPARTMENT_ID
 FROM HR.EMPLOYEES E, HR.DEPARTMENTS D
 WHERE E.DEPARTMENT_ID = D.DEPARTMENT_ID GROUP BY E.DEPARTMENT_ID;



/* PL/SQL Function Result Cache */
CREATE OR REPLACE FUNCTION get_name (id NUMBER) RETURN VARCHAR2
 RESULT_CACHE RELIES_ON(emp) IS ...

-- Package specification
CREATE OR REPLACE PACKAGE department_pks IS
 -- Function declaration
 FUNCTION get_dept_info (dept_id NUMBER) RETURN dept_info_record RESULT_CACHE;
END department_pks;

CREATE OR REPLACE PACKAGE BODY department_pks AS
 -- Function definition
 FUNCTION get_dept_info (dept_id NUMBER) RETURN dept_info_record
  RESULT_CACHE RELIES_ON (EMPLOYEES)
 IS
 BEGIN ...


/* Bypass Result Cache */
begin
 DBMS_RESULT_CACHE.BYPASS(TRUE);
 DBMS_RESULT_CACHE.FLUSH;
end;

exec DBMS_RESULT_CACHE.BYPASS(FALSE);



/* Monitoring Result Cache */
--  Memory taken for Result Cache
SELECT NAME, ROUND(BYTES/1024,2) KB
 FROM V$SGAstat
 WHERE upper(name) like '%RESULT%';
```

```
-- objects in the result cache
select ID, TYPE, STATUS, BUCKET_NO, HASH, NAME, NAMESPACE, CREATOR_UID,
DEPEND_COUNT, BLOCK_COUNT, SCN, COLUMN_COUNT, PIN_COUNT, SCAN_COUNT, ROW_COUNT,
ROW_SIZE_MAX, ROW_SIZE_MIN, ROW_SIZE_AVG, BUILD_TIME, LRU_NUMBER, OBJECT_NO,
INVALIDATIONS, SPACE_OVERHEAD, SPACE_UNUSED, CACHE_ID, CACHE_KEY,
to_char(CREATION_TIMESTAMP,'HH12:MI AM') CREATE_TIME
from   V$RESULT_CACHE_OBJECTS
order by type desc;

-- result cache stats
select ID, NAME, VALUE
from V$RESULT_CACHE_STATISTICS;

-- using DBMS_RESULT_CACHE
-- check the status of the Result Cache
--  Note: this is the reliable method to know whether result cache is enabled or
not
SQL>select DBMS_RESULT_CACHE.STATUS from dual;

-- display report on result cache memory
SQL>set serveroutput on
SQL>exec DBMS_RESULT_CACHE.MEMORY_REPORT

-- turn bypass mode on and off
SQL>exec DBMS_RESULT_CACHE.BYPASS (TRUE);

-- to flush the result cache
SQL>exec DBMS_RESULT_CACHE.FLUSH
```

Obtaining Information about Object Locks

```
-- DML locks by transactions on objects
SELECT l.object_id,
 l.session_id,
 l.oracle_username,
 DECODE(l.locked_mode,
  1, 'No Lock',
  2, 'Row Share',
  3, 'Row Exclusive',
  4, 'Shared Table',
  5, 'Shared Row Exclusive',
  6, 'Exclusive') locked_mode,
 (select o.object_name from  DBA_OBJECTS o WHERE o.object_id=l.object_id)
object_name
 FROM V$LOCKED_OBJECT l
 order by ORACLE_USERNAME;

SELECT dbl.lock_type, dbl.mode_held, dbl.blocking_others,
 dbo.object_name object_locked, dbo.object_type
FROM dba_locks dbl, v$session v, dba_objects dbo
WHERE v.username not in
('SYS','SYSTEM','DBSNMP','SYSMAN','OUTLN','TSMSYS','WMSYS','EXFSYS','CTXSYS','XDB'
,'ORDSYS','MDSYS','OLAPSYS','WKSYS','WK_TEST','IX')
AND dbl.session_id = v.sid AND dbo.object_id = dbl.lock_id1;
```

```
-- blocking session
select ADDR, KADDR, l.SID, s.USERNAME, S.MACHINE, S.TERMINAL, S.PROGRAM,
decode(l.TYPE, 'TM','DML enqueue','TX','Transaction enqueue','UL','User supplied')
Block_Type,
ID1, ID2, LMODE, REQUEST, CTIME, BLOCK
from v$lock l, v$session s
where s.sid = l.sid
and l.block=1;


-- blocking and blocked sessions
SELECT SUBSTR(TO_CHAR(w.session_id),1,5) WSID, p1.spid WPID,
SUBSTR(s1.username,1,12) "WAITING User",
SUBSTR(s1.osuser,1,8) "OS User",
SUBSTR(s1.program,1,20) "WAITING Program",
s1.client_info "WAITING Client",
SUBSTR(TO_CHAR(h.session_id),1,5) HSID, p2.spid HPID,
SUBSTR(s2.username,1,12) "HOLDING User",
SUBSTR(s2.osuser,1,8) "OS User",
SUBSTR(s2.program,1,20) "HOLDING Program",
s2.client_info "HOLDING Client",
o.object_name "HOLDING Object"
FROM gv$process p1, gv$process p2, gv$session s1,
gv$session s2, dba_locks w, dba_locks h, dba_objects o
WHERE -- w.last_convert > 120 AND (Objects locked for 2 mins)
    h.mode_held != 'None'
AND h.mode_held != 'Null'
AND w.mode_requested != 'None'
AND s1.row_wait_obj# = o.object_id
AND w.lock_type(+) = h.lock_type
AND w.lock_id1(+) = h.lock_id1
AND w.lock_id2 (+) = h.lock_id2
AND w.session_id = s1.sid (+)
AND h.session_id = s2.sid (+)
AND s1.paddr = p1.addr (+)
AND s2.paddr = p2.addr (+)
ORDER BY w.last_convert desc;
```

Handling a Hanging Database

- Archiver Process Stuck. In the alert log file, you'll see: ORA-00257: archiver error. Connect internal only, until freed.
    - Redirect archiving to a different directory.
    - Clear the archive log destination by removing some archive logs.
- Sys Audit is full. Check and delete $ORACLE_HOME/rdbms/audit
- Make sure the OS isn't swapping.

Accurately Measuring Process Size

- Some tools like ps and top give you a misleading idea as to the process size, because they include the common shared TEXT sizes in individual processes.

```
-- memory occupied by every session
SELECT sid, n.name|| ' ('||s.statistic#||')', to_char(round(value/1024),'999,999')
|| ' KB' KB
FROM v$sesstat s, v$statname n
WHERE s.statistic# = n.statistic#
AND n.name like '%ga memory%'
ORDER BY SID, value DESC;


-- total memory allocated to the PGA and UGA memory
SELECT 'Total PGA', round(SUM(value)/1024/1024,2) MB
FROM V$SESSTAT s, V$STATNAME n
WHERE s.statistic# = n.statistic#
AND n.name in ('session pga memory')
group by n.name
union
SELECT 'Total UGA', round(SUM(value)/1024/1024,2) MB
FROM V$SESSTAT s, V$STATNAME n
WHERE s.statistic# = n.statistic#
AND n.name in ('session uga memory')
group by n.name;


-- another way to know PGA used size
select round(value/1024/1024,2) MB from V$PGASTAT where name='total PGA inuse';
```

Managing Automatic Workload Repository (AWR)

- The AWR and ADDM are Oracle products that need special licensing through the purchase of the Diagnostic Pack.

```
-- To manually creating a snapshot:
dbms_workload_repository.create_snapshot()


-- to list snapshots taken by AWR
select SNAP_ID, STARTUP_TIME, BEGIN_INTERVAL_TIME, END_INTERVAL_TIME,
FLUSH_ELAPSED, SNAP_LEVEL, ERROR_COUNT
from DBA_HIST_SNAPSHOT
order by STARTUP_TIME DESC, SNAP_ID DESC


-- To drop a range of snapshots:
dbms_workload_repository.drop_snapshot_range (low_snap_id => 40,high_snap_id =>
60, dbid => 2210828132)

-- To modify a AWR setting:
begin
DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS(
 retention => 43200, -- in mins
 interval => 30,     -- between snaps in mins
 dbid => 3310949047  -- If NULL, the local dbid will be used
  );
end;
/

-- To get AWR setting
select DBID, SNAP_INTERVAL, RETENTION
from DBA_HIST_WR_CONTROL;



/* Creating and Deleting AWR Snapshot Baselines */
begin
dbms_workload_repository.create_baseline
 (start_snap_id => 125,
  end_snap_id => 185,
  baseline_name => 'peak_time baseline',
  dbid => 2210828132 );
end;
/

-- to list created baselines:
select DBID, BASELINE_ID, BASELINE_NAME, START_SNAP_ID, START_SNAP_TIME,
END_SNAP_ID, END_SNAP_TIME
from DBA_HIST_BASELINE
order by START_SNAP_ID;

-- To drop a snapshot baseline:
--  if cascade is true, the snaps will also be deleted
exec dbms_workload_repository.drop_baseline (baseline_name => 'peak_time
baseline', cascade => FALSE)
```

```
/* Creating AWR Reports */
<ORACLE_HOME>\RDBMS\ADMIN\awrrpt.sql
-- allows you to select a single instance:
<ORACLE_HOME>\RDBMS\ADMIN\awrrpti.sql
```

Managing Automated Maintenance Tasks

- Automatic Optimizer Statistics Collection
- Automatic Segment Advisor
- Automatic SQL Tuning Advisor

```
/* Obtain Info about AMTs (11g) */

select *
from DBA_AUTOTASK_TASK


/* Monitoring AMT */
SELECT client_name, status,
 attributes, window_group,service_name
 FROM dba_autotask_client;



/* Enabling and Disableing a Maintenance Task */
begin
 DBMS_AUTO_TASK_ADMIN.ENABLE
 (client_name => 'sql tuning advisor',
  operation => 'automatic sql tuning task',
  window_name => 'monday_window');
 end;
/

begin
 dbms_auto_task_admin.disable
 (client_name => 'sql tuning advisor',
  operation => 'automatic sql tuning task',
  window_name => 'monday_window');
 end;
/
```

Using Automatic Database Diagnostic Monitor (ADDM)

- The AWR and ADDM are Oracle products that need special licensing through the purchase of the Diagnostic Pack.

```
/* Obtain info about ADDM */
-- list of all advisors tasks
select * from DBA_ADVISOR_TASKS;

-- findings (all advisors)
select * from DBA_ADVISOR_FINDINGS where task_name='ADDM 01012009';

-- findings only add ADDM
select * from DBA_ADDM_FINDINGS where task_name='ADDM 01012009';

-- recommendations
select * from DBA_ADVISOR_RECOMMENDATIONS where task_name='ADDM 01012009';
```

```
/* Setting Up ADDM */
-- CONTROL_MANAGEMENT_PACK_ACCESS should be DIAGNOSTIC or DIAGNOSTIC+TUNING
-- STATISTICS_LEVEL should be TYPICAL or ALL (not BASIC)
select name, value
from v$parameter where upper(name) in
('CONTROL_MANAGEMENT_PACK_ACCESS','STATISTICS_LEVEL');

alter system set control_management_pack_access='DIAGNOSTIC+TUNING'



/* Determining Optimal I/O Performance */
-- Oracle assumes the value of the parameter (not intialization parameter)
--  DBIO_EXPECTED is 10 milliseconds.
SELECT PARAMETER_VALUE
FROM DBA_ADVISOR_DEF_PARAMETERS
WHERE ADVISOR_NAME='ADDM'
AND PARAMETER_NAME='DBIO_EXPECTED'
If your hardware is significantly different, you can set the parameter value one
time for all subsequent ADDM executions:
DBMS_ADVISOR.SET_DEFAULT_TASK_PARAMETER('ADDM','DBIO_EXPECTED', 8000);



/* Retreiving AWR snapshots */
SELECT SNAP_ID , INSTANCE_NUMBER ,
 TO_CHAR(BEGIN_INTERVAL_TIME,'DD-MON-RR HH24:MI') BEGIN_TIME,
 TO_CHAR(END_INTERVAL_TIME,'DD-MON-RR HH24:MI') END_TIME
FROM DBA_HIST_SNAPSHOT
order by SNAP_ID desc;



/* Running ADDM in Database Mode */
DECLARE
 -- task name shouldn't exists in db
 TNAME VARCHAR2(50) :='ADDM DB 01012009';
BEGIN
DBMS_ADDM.ANALYZE_DB (
 task_name      =>TNAME, -- it is IN OUT para
 begin_snapshot =>43,
 end_snapshot   =>44);
END;
/

-- also the report will then be generated
@C:\oracle\oracledb11g\RDBMS\ADMIN\addmrpt.sql



/* Running ADDM in Instance Mode */
-- used in RAC
DECLARE
 TNAME VARCHAR2(50) :='ADDM INST 01012009';
BEGIN
DBMS_ADDM.ANALYZE_INST (
```

```
 task_name      =>TNAME, -- it is IN OUT para
 begin_snapshot =>43,
 end_snapshot   =>44,
 instance_number=>1);
END;
/



/* Running ADDM in Partial Mode */
-- subset of all database instances (in RAC)
DECLARE
 -- task name shouldn't exists in db
 TNAME VARCHAR2(50) :='ADDM PART 01012009';
BEGIN
DBMS_ADDM.ANALYZE_PARTIAL (
 task_name      =>TNAME,
 begin_snapshot =>43,
 end_snapshot   =>44,
 instance_numbers=>'1,2,3');
END;
/



/* Displaying an ADDM Report */
SET LONG 1000000 PAGESIZE 0;
SELECT DBMS_ADDM.GET_REPORT('ADDM DB 01012009') FROM DUAL;



/* Deleting already created tasks */
exec DBMS_ADDM.DELETE(task_name=>'ADDM INST 01012009');
```

Using Automatic SQL Tuning Advisor

When SQL Tunining Advisor is running as an automatic task, it avoids analysing the following:

- Parallel queries
- Ad hoc queries
- Recursive statements
- SQL statements that use the INSERT and DELETE statements
- SQL statements that use DDL statements such as CREATE TABLE AS SELECT

```
/* Obtain Info about Automatic SQL Tuning Advisor */
select OWNER, TASK_ID, TASK_NAME, EXECUTION_NAME, DESCRIPTION,
 EXECUTION_TYPE, EXECUTION_TYPE#, EXECUTION_START, EXECUTION_END,
 ADVISOR_NAME, STATUS, STATUS_MESSAGE, ERROR_MESSAGE
from DBA_ADVISOR_EXECUTIONS
order by task_id desc

-- all SQL compilation and execution statistics
DBA_ADVISOR_SQLSTATS
```

```
DBA_ADVISOR_SQLPLANS



/* Configuring Automatic SQL Tuning */
-- Automatic SQL Tuning Advisor job runs for a maximum of one hour by default
-- in seconds
exec dbms_sqltune.set_tuning_task_parameter ('SYS_AUTO _SQL_TUNING_TASK',
'TIME_LIMIT', 14400)

-- other parameters to set
ACCEPT_SQL_PROFILES: TRUE/FALSE whether the database must automatically accept a
SQL profile
MAX_AUTO_SQL_PROFILES: maximum number of automatic SQL profiles allowed on the
system, in sum
MAX_SQL_PROFILES_PER_EXEC: maximum number of SQL profiles that can be
automatically implemented per execution of the task.

-- to view current parameter values:
COLUMN parameter_value FORMAT A30
SELECT parameter_name, parameter_value
    FROM   dba_advisor_parameters
    WHERE  task_name = 'SYS_AUTO_SQL_TUNING_TASK'
    AND    parameter_name IN ('ACCEPT_SQL_PROFILES',
                              'MAX_SQL_PROFILES_PER_EXEC',
                              'MAX_AUTO_SQL_PROFILES');



/* Enabling and Disable the automatic advisor */
-- enable it in all maintenance windows
begin
 dbms_auto_task_admin.enable (
  client_name => 'sql tuning advisor',
  operation => 'NULL',
  window_name='NULL');
end;
/

-- in a specific window
begin
dbms_auto_task_admin.enable (
 client_name => 'sql tuning advisor',
 operation => 'NULL',
 window_name='monday_night_window');
end;
/



/* View the Report */
-- display report of most recent execution:
VARIABLE l_report CLOB;
BEGIN
  :l_report := DBMS_SQLTUNE.report_auto_tuning_task(
    begin_exec   => NULL,
    end_exec     => NULL,
    type         => DBMS_SQLTUNE.type_text,     -- 'TEXT'
```

```
    level        => DBMS_SQLTUNE.level_typical, -- 'TYPICAL'
    section      => DBMS_SQLTUNE.section_all,   -- 'ALL'
    object_id    => NULL,
    result_limit => NULL);
END;
/
print :l_report
```

Implementing Automatic Memory Management

When you configure the database to use MEMORY_TARGET, you should take into consideration the following:

- The parameter STATISTICS_LEVEL must be set to TYPICAL
- If you set the parameters SGA_TARGET and PGA_TARGET, Oracle will consider the values as the minimum values for SGA and PGA.

Note You may set PGA_AGGREGATE_TARGET to a value and then a command issued that is heavy on PGA, Oracle will try to expand the actual PGA size to serve the command. If Oracle could not find enough free memory in the machine, the following error will be returned:

```
ORA-04030: out of process memory when trying to allocate 8192 bytes (sort subheap,sort key)
```

Resolving the issue can be done by reducing the PGA to a lower value. Also, high value of SORT_AREA_SIZE may cause it. Consider reducing its value (to something like 1MB).

- If you do not set the parameters SGA_TARGET and PGA_TARGET (or set them to zero), no minimum value is considered by Oracle for the SGA and PGA. When the instance starts, it assigns 60 percent to SGA and 40 percent to the PGA.
- When MEMORY_TARGET is configured, the following components are auto tuned: DB BUFFER CACHE, SHARED POOL, JAVA POOL, LARGE POOL and STREAMS POOL.

Note On Linux systems, if you receive the following error after setting the `MEMORY_TARGET` parameter, most likely the reason is that the /dev/shm is allocated a size (can be known by issuing the command `df -k`) less than `SGA_MAX_SIZE`:

```
ORA-00845: MEMORY_TARGET not supported on this system.
```

Resolving the issue can be done by the following OS commands:

```
#umount  /dev/shm
#mount -t tmpfs shmfs -o *size=><xx>m* /dev/shm
```

```
show parameter MEMORY_TARGET
show parameter MEMORY_MAX_SIZE

-- dynamic parameter
ALTER SYSTEM SET MEMORY_TARGET = 410M ;


-- To set a proper value to the parameter MEMORY_TARGET (11g):
V$MEMORY_TARGET_ADVICE.
SELECT * FROM V$MEMORY_TARGET_ADVICE order by MEMORY_SIZE desc;


-- To display current status of the memory components, use the following
query:
col COMPONENT format a30
SELECT COMPONENT, ROUND(CURRENT_SIZE/1024/1024) CURRENT_SIZE ,
ROUND(MIN_SIZE/1024/1024) MIN, ROUND(MAX_SIZE/1024/1024) MAX
FROM V$MEMORY_DYNAMIC_COMPONENTS
order by CURRENT_SIZE desc;
```

```
-- To know how Oracle has modified the memory area sizes by time (11g):
select START_TIME, END_TIME, STATUS, COMPONENT, OPER_TYPE, OPER_MODE,
 PARAMETER, INITIAL_SIZE/1024/1024 INITIAL_SIZE_MB,
 TARGET_SIZE/1024/1024 TARGET_SIZE_MB, FINAL_SIZE/1024/1024 FINAL_SIZE_MB
from V$MEMORY_RESIZE_OPS
order by START_TIME, END_TIME
```

Configuring DB_nK_CACHE_SIZE

```
ALTER SYSTEM SET DB_16K_CACHE_SIZE =1024M;

CREATE TABLESPACE big_block_tbs
 DATAFILE '/test01/app/oracle/big_block_01.dbf' SIZE 1000M
 BLOCKSIZE 16K;

ALTER TABLE names2 MOVE TABLESPACE big_block_tbs;
```

Managing Optimizer Operations

Setting the Optimizer Mode

```
OPTIMIZER_MODE:
ALL_ROWS (= CHOOSE)maximum throughput (good for patch processing)
FIRST_ROWS_n where n 1,10,100,1000 maximum response time (good for interactive
apps)
FIRST_ROWS (deprecated) same as above

SHOW PARAMETER OPTIMIZER_MODE

select value from v$parameter where upper(name)='OPTIMIZER_MODE' ;


-- instance level:
ALTER SYSTEM SET optimizer_mode = first_rows_10;

-- session level (PL/SQL blocks won't be affected)
ALTER SESSION SET optimizer_mode = first_rows_10;


-- statement level
select /*+ first_rows(10) */  ..
SELECT  /*+ ALL_ROWS */  ...
```

Defining Access Paths and Joins for the Query Optimizer

Access Paths:

- Full Table Scans (FULL)
- Rowid Scans
- Index Scans:
    - Index Unique Scans (INDEX): using UNIQUE or PRIMARY KEY indexes
    - Index Range Scans (INDEX, INDEX_ASC, INDEX_DESC, NO_INDEX_RS): <,>,=, Like on index key
    - Index Range Scans Descending (INDEX_DESC): when an order by descending clause can be satisfied by an index
    - Index Skip Scans (INDEX_SS, INDEX_SS_ASC, INDEX_SS_DESC): when there are few distinct values in the leading column of the composite index and many distinct values in the nonleading key of the index. For example: INDEX(sex, emp_id) and select ... where emp_id=100
    - Full Scans: it is available if a predicate references one of the columns in the index or all of the columns in the table referenced in the query are included in the index. It read single blocks.
    - Fast Full Index Scans (INDEX_FFS, INDEX_FFS_ASC, INDEX_FFS_DESC, NO_INDEX_FFS): when the index contains all the columns that are needed for the query, and at least one column in the index key has the NOT NULL constraint. It reads the entire index using multiblock reads, unlike a full index scan, and can be parallelized.

    - Index Joins (INDEX_JOIN): it is a hash join of several indexes that together contain all the table columns that are referenced in the query.
    - Bitmap Indexes (INDEX_COMBINE or INDEX): the optimizer uses a bitmap index to get the rowids. It requires EE license.
- Cluster Access (CLUSTER): when a table is stored in an indexed cluster.
- Hash Access (HASH): is used to locate rows in a hash cluster, based on a hash value.
- Sample Table Scans: when using SAMPLE [BLOCK] keyword in the select statement.

Joins:

- Nested Loop Join (USE_NL, No_USE_NL): for every row in the outer table, Oracle accesses all the rows in the inner table.
- Hash Join (USE_HASH, NO_USE_HASH): used when joining one large table with a small one. The smaller data source is built into memory as hash table. It then scans the larger table, probing the hash table to find the joined rows. Euality condition is a must.
- Sort Merge (USER_MERGE, NO_USE_MERGE): It performs well when no order is required on the data source (or already sorted) and the join condition is NOT equality.
- Outer Joins: extends the result of a simple join:
    - Nested Loop Outer Joins: The outer table will be the one with rows that are being preserved.
    - Hash Join Outer Joins
    - Sot Merge Outer Joins
    - Full Outer Join
- Cartesian: two or more tables with no joins.

```
/* Access Path Hints */
-- Full Table Scans
select /* FULL(table_alias) */

-- "order by" might be served by the index
select /*+ index_DESC(n names3i2) */
*
from names3 n
where name like '%JZ'
order by NAME DESC

-- you can indicate whether to save the block in buffer cache
select /* CACHE .
select /* NOCACHE .



-- Index Unique Scan Hints
-- Index Unique Scan as access path is auto used by the optimizer
-- you can define the hint to use the index though:
select /* INDEX(table_alias index_name) */ ...
select /*+ INDEX(e1 emp_emp_id_pk) .. */ from employees e1, ..

SELECT /*+ index(t1 t1_abc) index(t2 t2_abc) */ COUNT(*)
FROM t1, t2
WHERE t1.col1 = t2.col1;
```

```
-- Index Range Scan Hints
-- to instruct the optimizer to use a specific index (to avoid fts):
select /* INDEX(table_alias index_name) */ ...
select /*+ INDEX(e1 emp_emp_id_pk) .. */ from employees e1, ..



-- Index Range Scan Descending Hints
-- use it when an order by descending clause can be satisfied by an index
select /*+ INDEX_DESC(t indexC) */
 * from try t where C < sysdate order by C desc

-- exclude the index range scan access path
SELECT /*+ NO_INDEX_RS(e emp_name_ix) */ last_name
FROM employees e
WHERE first_name BETWEEN 'A' AND 'B';


-- Fast Full Index Scan Hints
select /*+ INDEX_FFS(t indexb) */
 b
from try t
where t.b='b'

select /*+ index_ffs_asc(n i1) */
name
from names2 n
order by name

SELECT /*+ NO_INDEX_FFS(i pk_serv_inst) NO_INDEX_FFS(i ix_serv_inst) */
latitude
FROM servers s, serv_inst i
WHERE s.srvr_id = i.srvr_id;



-- Index Join Hints
SELECT /*+ INDEX_JOIN(e emp_manager_ix emp_department_ix) */ department_id
FROM employees e
 WHERE manager_id < 110
AND department_id < 50;



-- Bitmap Index
-- bitmap indexes should be there
SELECT  /*+ INDEX_COMBINE(e emp_manager_ix emp_department_ix) */ *
FROM employees e
WHERE (manager_id = 108) OR (department_id = 110);



-- Cluster
CREATE CLUSTER sc_srvr_id (
srvr_id NUMBER(10)) SIZE 1024;
```

```
CREATE INDEX idx_sc_srvr_id ON CLUSTER sc_srvr_id;

CREATE TABLE cservers
CLUSTER sc_srvr_id (srvr_id) AS SELECT * FROM servers;

CREATE TABLE cserv_inst
CLUSTER sc_srvr_id (srvr_id) AS SELECT * FROM serv_inst;

SELECT /*+ CLUSTER(cservers) */ srvr_id
FROM cservers
WHERE srvr_id = 503
GROUP BY srvr_id;



-- Hash
CREATE CLUSTER sthc_si (srvr_id NUMBER(10))
SIZE 1024 SINGLE TABLE HASHKEYS 11
TABLESPACE uwdata;

CREATE TABLE si_hash
CLUSTER sthc_si (srvr_id) AS
SELECT *
FROM serv_inst;

SELECT /*+ HASH(si_hash) */ srvr_id
FROM si_hash
WHERE srvr_id = 503
GROUP BY srvr_id;


-- Sample data
-- SAMPLE (1) : read 1% of rows
-- SAMPLE BLOCK (1) : read 1% of blocks
SELECT * FROM employees SAMPLE BLOCK (1);



/* Join Operations Hints */

-- Nested Loops
select /*+ use_nl(e d) use_nl(e j) */ -- or No_use_nl
employee_id , department_name, salary, j.job_title
from employees e , departments d, jobs j
where e.department_id = d.department_id
  and e.job_id = j.job_id
  and d.department_id>10
/


-- Hash Join
SELECT /*+ USE_HASH(o l) */
 o.customer_id, l.unit_price * l.quantity
FROM orders o ,order_items l
WHERE l.order_id = o.order_id;
```

```
-- Sort Merge
select /*+ use_merge(e g) */
employee_id , last_name, grade
from employees e, grades g
where e.salary between g.ssal and g.esal;


-- Full Outer Join
SELECT d.department_id, e.employee_id
FROM employees e
FULL OUTER JOIN departments d
 ON e.department_id = d.department_id
ORDER BY d.department_id;
```

Gathering Optimizer Statistics

Gathering Object Statistics

- See Collecting Object Statistics. Preferences can be set. See next section.
- Collecting Dictionary Object Stats, see below. Recommended every week in off beak times. It may take considerable time (hours).

```
/* Retreive Gathered Stats */
-- table stats
select OWNER, TABLE_NAME, PARTITION_NAME, PARTITION_POSITION,
SUBPARTITION_NAME, SUBPARTITION_POSITION, OBJECT_TYPE, NUM_ROWS, BLOCKS,
EMPTY_BLOCKS, AVG_SPACE, CHAIN_CNT, AVG_ROW_LEN, AVG_SPACE_FREELIST_BLOCKS,
NUM_FREELIST_BLOCKS, AVG_CACHED_BLOCKS, AVG_CACHE_HIT_RATIO, SAMPLE_SIZE,
LAST_ANALYZED, GLOBAL_STATS, USER_STATS, STATTYPE_LOCKED, STALE_STATS
from DBA_TAB_STATISTICS
where owner not in
('SYS','SYSTEM','DBSNMP','SYSMAN','OUTLN','TSMSYS','WMSYS','EXFSYS','CTXSYS','
XDB','ORDSYS','MDSYS','OLAPSYS','WKSYS','WK_TEST','IX')
  and owner NOT LIKE 'FLOWS_%'
order by OWNER, LAST_ANALYZED DESC;

-- col stats
select OWNER, TABLE_NAME, COLUMN_NAME, NUM_DISTINCT, LOW_VALUE, HIGH_VALUE,
DENSITY, NUM_NULLS, NUM_BUCKETS, LAST_ANALYZED, SAMPLE_SIZE, GLOBAL_STATS,
USER_STATS, AVG_COL_LEN, HISTOGRAM
from DBA_TAB_COL_STATISTICS
where owner not in
('SYS','SYSTEM','DBSNMP','SYSMAN','OUTLN','TSMSYS','WMSYS','EXFSYS','CTXSYS','
XDB','ORDSYS','MDSYS','OLAPSYS','WKSYS','WK_TEST','IX')
  and owner NOT LIKE 'FLOWS_%'
order by OWNER, LAST_ANALYZED DESC;



/* Collecting data dicionary stats */
-- for fixed tables
conn / as sysdba
EXECUTE DBMS_STATS.GATHER_FIXED_OBJECTS_STATS

-- for real dictionary tables:
-- SYS and SYSTEM users as well as the owners of all database components
```

```
conn / as sysdba
-- it may take hours to finish
EXECUTE DBMS_STATS.GATHER_DICTIONARY_STATS
```

Gathering System Statistics

- Gathering mode controlled by GATHERING_MODE parameter:
  - No-Workload Mode: implemented by passed NOWORKLOAD. It gathers general I/O stats.
  - Workload Mode: START will manually start the gathering. STOP will manually stops it. INTERVAL gathering is controlled by INTERVAL parameter.
- See Managing OS statistics.

```
-- start gathering
EXECUTE dbms_stats.gather_system_stats('start');


-- stop gathering and save the stats
EXECUTE dbms_stats.gather_system_stats('stop');


-- for three minutes interval
EXECUTE dbms_stats.gather_system_stats('INTERVAL',3);


-- view gathered stats
SELECT
  SNAME,
  PNAME,
  decode(PNAME,
  'IOTFRSPEED','I/O transfer Speed (B/ms)',
  'IOSEEKTIM','Seek+Latency+OS Overhead Time (ms)',
  'SREADTIM','Single-Block Read Average Time (ms)',
  'MREADTIM','MBRC Block Sequential Average Read Time (ms)',
  'CPUSPEED','Average Number of CPU Cycles Captured for the Workload',
  'CPUSPEEDNW','Average Number of CPU Cycles Captured for the Non-Workload',
  'MBR','Average Multiblock Read Count for Sequential read (in Blocks)',
  'MAXTHR','Maximum I/O System Throughput (B/s)',
  'SLAVETHR','Average Slave I/O Throughput (B/s)',
  PNAME) Description,
  PVAL1,
  PVAL2
FROM sys.aux_stats$;
```

Changing Statistics Preferences

The function DBMS_STATS.GET_PARAM is used to in Oracle 10g to return the default values of parameters of the DBMS_STATS package. This function is now obsolete in Oracle 11g and replaced with GET_PREFS procedure. Following is an example:

```
SET SERVEROUTPUT ON
declare
 v_value varchar2(100);
begin
   v_value := DBMS_STATS.GET_PREFS (
     PNAME =>'STALE_PERCENT',
```

```
     OWNNAME =>'HR',
     TABNAME =>'EMPLOYEES');
 DBMS_OUPTPUT.PUT_LINE(v_value);
end;
```

Regarding the GET_PREFS function, consider the following:

- PNAME parameter indicates the preference name and can take one of the following values: CASCADE, DEGREE, ESTIMATE_PERCENT, METHOD_OPT, NO_INVALIDATE, GRANULARITY, PUBLISH, INCREMENTAL and STALE_PERCENT.
- If the OWNNAME and TABNAME are provided and a preference has been entered for the table, the function returns the preference as specified for the table. In all other cases it returns the global preference if it has been specified, otherwise the default value is returned.

SET_GLOBAL_PREFS, SET_DATABASE_PREFS, SET_SCHEMA_PREFS, SET_TABLE_PREFS procedures are used to set the statistics preferences for the global, database, schema or table levels respectively. Following is an example:

```
begin
 DBMS_STATS.SET_GLOBAL_PREFS ( PNAME =>'ESTIMATE_PERCENT', PVALUE =>'75');
end;
```

Similarly, the procedures DELETE_*_PREFS are used to delete current statistics preferences. EXPORT_*_PREFS and IMPORT_*_PREFS procedures are used to export and import statistics preferences. Following is an example:

```
begin
 DBMS_STATS.EXPORT_DATABASE_PREFS(
  STATTAB =>'mytable',    -- table name to where statistics should be exported
  STATID  =>'prod_prefs', -- identifier to associate with these statistics
  STATOWN =>'HR');        -- Schema containing stattab (if other than ownname)
end;
```

Managing Pending and Published Statistics

Starting with Oracle 11g, when gathering statistics, you have the option to automatically publish the statistics at the end of the gather operation (default behavior), or to have the new statistics saved as pending. Saving the new statistics as pending allows you to validate the new statistics and publish them only if they are satisfactory.

You can check whether or not the statistics will be automatically published checking the value of the PUBLISH attribute using the DBMS_STATS package as in the following example:

```
SELECT DBMS_STATS.GET_PREFS('PUBLISH') PUBLISH FROM DUAL;
```

You can change the PUBLISH setting at either the schema or table level. Following are examples to do so:

```
-- setting PUBLISH at schema level
exec DBMS_STATS.SET_SCHEMA_PREFS('HR', 'PUBLISH', 'FALSE');

-- setting PUBLISH at table level
exec DBMS_STATS.SET_TABLE_PREFS('HR','EMPLOYEES', 'PUBLISH', 'FALSE');
```

Published statistics are stored in data dictionary views, such as DBA_TAB_STATISTICS and USER_IND_STATISTICS. Pending statistics are stored in views such as DBA_TAB_PENDING_STATISTICS and DBA_IND_PENDING_STATISTICS.

```
select NUM_ROWS, BLOCKS, AVG_ROW_LEN, SAMPLE_SIZE, LAST_ANALYZED
from   DBA_TAB_PENDING_STATISTICS where  OWNER='HR' AND TABLE_NAME ='EMPLOYEES';
```

By default, the optimizer uses the published statistics stored in the data dictionary views. If you want the optimizer to use the newly collected pending statistics, set the initialization parameter

OPTIMIZER_PENDING_STATISTICS to TRUE (the default value is FALSE), and then run a workload against the table or schema or just gather its statistics:

```
ALTER SESSION SET OPTIMIZER_PENDING_STATISTICS = TRUE;
```

The optimizer will use the pending statistics (if available) instead of the published statistics when compiling SQL statements. If the pending statistics are valid, they can be made public by executing the following statement:

```
-- for the whole database
exec DBMS_STATS.PUBLISH_PENDING_STATS(null, null);

-- publishing specific database object pending statistics
exec DBMS_STATS.PUBLISH_PENDING_STATS('HR','EMPLOYEES');
```

If you do not want to publish the pending statistics, delete them by executing the following statement:

```
exec DBMS_STATS.DELETE_PENDING_STATS('HR','EMPLOYEES');
```

Managing Extended Statistics

MultiColumn Statistics

When multiple columns from a single table are used together in the where clause of a query (multiple single column predicates), Oracle optimizer in previous versions (before 11g) does not understand the relationship between the columns. In Oracle 11g, statistics on these columns as a group (column group) can be gathered and thus resulting in high enhancement in CBO cardinality estimation.

You can also create column groups manually by using the CREATE_EXTENDED_STATS function in the DBMS_STATS package. This function creates a column statistics entry in the system for a user specified column group or an expression in a table. This function returns a system-generated name of this newly created entry for the extension.

Note: The optimizer will only use MultiColumn statistics with equality predicates.

Following is an example:

```
declare
 V_NAME VARCHAR2(30);
begin
 -- stats of the combined columns will be collected next time the stats is gathered
 V_NAME := DBMS_STATS.CREATE_EXTENDED_STATS(
           OWNNAME   => NULL,
           TABNAME   => 'EMPLOYEES',
           EXTENSION => '(STATE_ID,COUNTRY_ID)');

 -- you can then issue the gathering process
 DBMS_STATS.GATHER_TABLE_STATS  (NULL,  'EMPLOYEES',  METHOD_OPT='for  all  columns  size
skewonly' );
end;
```

The DBMS_STATS.GATHER_TABLE_STATS procedure can also be used to create column group and gather its statistics all in one step. The keyword FOR COLUMNS is used in this case as shown in the following example:

```
begin
 DBMS_STATS.GATHER_TABLE_STATS ('HR', 'EMPLOYEES',
  METHOD_OPT=>'for all columns size skewonly for columns (STATE_ID,COUNTRY_ID)' );
end;
```

Note The default value of METHOD_OPT is 'FOR ALL COLUMNS SIZE AUTO' which makes Oracle create column groups for a table, based on the workload analysis, similar to how it is done for histograms.

You can use the methods in the following code examples to retrieve information on column groups that have been created:

```
-- you can query the data dictionary USER_STAT_EXTENSIONS
select EXTENSION_NAME, EXTENSION from USER_STAT_EXTENSIONS where TABLE_NAME='EMPLOYEES';

-- you can query USER_TAB_COL_STATISTICS (extension name appears as COLUMN_NAME)
select COLUMN_NAME, NUM_DISTINCT, HISTOGRAM
from USER_TAB_COL_STATISTICS where TABLE_NAME = 'EMPLOYEES';

-- you can use DBMS_STATS.SHOW_EXTENDED_STATS_NAME function
select DBMS_STATS.SHOW_EXTENDED_STATS_NAME(OWNNAME   => 'HR',
                                           TABNAME   => 'EMPLOYEES',
                                           EXTENSION => 'STATE_ID,COUNTRY_ID') AS E_NAME
from dual;
```

After gathering the multi-column statistics as show in the example, when you check the explain plan for a query of a where condition like "STATE_ID = 'CA' AND COUNTRY_ID = 'US'", you will notice that the optimizer has retrieved the correct number of expected retrieved rows. Practically, this will lead to a significant improvement in the statement execution.

Following is how to drop a column group that you have previously defined:

```
exec DBMS_STATS.DROP_EXTENDED_STATS('HR', 'EMPLOYEES', '(STATE_ID,COUNTRY_ID)');
```

Expression Statistics

In Oracle 11g, you can create statistics on an expression. Following are examples to do that:

```
declare
 V_NAME VARCHAR2(30);
begin
 -- to create expression extended stats (not statistics are yet gathered)
 V_NAME := DBMS_STATS.CREATE_EXTENDED_STATS(
            OWNNAME   => NULL,
            TABNAME   => 'EMPLOYEES',
            EXTENSION => '(lower(last_name))');
end;

begin
 -- to create expression extended stats and gather the statistics in one step
 DBMS_STATS.GATHER_TABLE_STATS(OWNNAME=>NULL,TABLE_NAME=>'EMPLOYEES',
 METHOD_OPT=>'for all columns size skewonly for columns (lower(last_name))');
end;
```

After executing the code above, if you issue a query from EMPLOYEES table with a condition like LOWER(LAST_NAME)='ABC', the optimizer has statistics about the retrieved rows and thus will be able to estimate the correct number of returned rows. Consequently, the optimizer will most likely create a more efficient plan than if those statistics were not present.

Use the DBA_STAT_EXTENSIONS data dictionary view to retrieve information on expression statistics that have been created in the database.

```
select EXTENSION_NAME, EXTENSION from USER_STAT_EXTENSIONS where TABLE_NAME='EMPLOYEES';
```

Following is an example of the removal of an extended expression statistic:

```
exec DBMS_STATS.DROP_EXTENDED_STATS(null, 'EMPLOYEES', '(lower(lat_name))' );
```

Note that you will not be able to drop an extended expression statistics, if a function-based index is dependent on that statistic (ORA-20000 error will be returned).

A Simple Approach to Tuning SQL Statements

- Identify Problem Statements
  - Using V$SQL or SQL Tracing
  - AWR report and the ADDM analysis
- Locate the Source of the Inefficiency
  - Show the EXPLAIN PLAN of the statement
  - Use the SQL Access Advisor and SQL Tunining Advisor

Using Application Tracing Tools

Using the SQL Trace Facility and TKPROF

```
1. Set init parameters for trace file management.
They are all system and session level
STATISTICS_LEVEL =  Typical or ALL
Even if STATISTICS_LEVEL is set to TYPICAL or ALL, you can keep the database
from tracing by
using:
ALTER SESSION statement to set TIMED_STATISTICS to FALSE.

show parameter USER_DUMP_DEST

show parameter MAX_DUMP_FILE_SIZE

2. Enable the SQL Trace facility for the desired session.
-- current session
ALTER SESSION SET sql_trace=true;

begin
 sys.dbms_session.set_sql_trace(true);
end;
/


-- specific session
exec DBMS_SYSTEM.SET_SQL_TRACE_IN_SESSION(SID=>123, SERIAL#=>567, SQL_TRACE=>
true);
exec DBMS_MONITOR.SESSION_TRACE_ENABLE(session_id => 27, serial_num => 60,
waits => TRUE, binds => FALSE);


3. Run TKPROF to translate the trace file into a readable output file and
 optionally create a SQL script that can be used to store the statistics in a
database.
tkprof finance_ora_16340.trc test.txt sys=no explain=y AGGREGATE =yes|no WAITS
=yes|no


4. Interpret the output file.
Row Source Operations
 cr specifies consistent reads performed by the row source
 specifies physical reads performed by the row source
 w specifies physical writes performed by the row source
 time specifies time in microseconds
```

Using the Event 10046 to Trace SQL Code

- This method is more advanced than the SQL Trace utility.

```
alter session set timed_statistics = true;
alter session set statistics_level=all;
alter session set max_dump_file_size=unlimited;
alter session set events '10046 trace name context forever, level 12';
```

```
… run the code to trace
ALTER SESSION SET EVENT='10046 trace name context off';
-- trace file will be in USER_DUMP_DEST (usually udump folder)

-- not recommended
ALTER SYSTEM SET EVENT='10046 trace name context forever, level 4'
SCOPE=spfile;
ALTER SYSTEM SET EVENT='' SCOPE=spfile;


-- Setting the 10046 Event For a Session Other than Your Own
-- Method 1
-- SPID known (OR next)
oradebug setospid 1864;
-- PID known
oradebug setpid 21;

-- then
oradebug event 10046 trace name context forever, level 12
oradebug event 10046 trace name context off

-- Method 2
Dbms_system.set_ev (
si binary_integer, -- SID
se binary_integer, -- Serial#
ev binary_integer, -- Event code or number to set.
le binary_integer, -- Usually level to trace
cm binary_integer -- When to trigger (NULL = context forever.)
```

Tracing End to End Application

```
/* To specify module and action names */
exec
DBMS_APPLICATION_INFO.SET_MODULE(MODULE_NAME=>'FINANCIAL',ACTION_NAME=>'PAYROL
L');
exec DBMS_APPLICATION_INFO.SET_ACTION(ACTION_NAME=>'PAY SLIP');



/* To add your own marker to the trace file names */
-- so you can more easily find the generated files.
ALTER SESSION SET TRACEFILE_IDENTIFIER ="fin_payrol";



/* Enabling Collection of Client and Service Statistics */
For client-Level Statistics use:
DBMS_MONITOR.CLIENT_ID_STAT_ENABLE(<client_id>)

For Service-Level Statistics:
SELECT NAME FROM V$ACTIVE_SERVICES ;
select service_name from v$session where username='HR';

-- service name case sensitive
DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE(<service_name>,<module_name>,
<action_name>)
```

```
For example:
DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE( service_name=>'APPS1',module_name
=>'PAYROLL')

To enable tracing for a Service named APPS1:
DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE('APPS1',DBMS_MONITOR.ALL_MODULES,
DBMS_MONITOR.ALL_ACTIONS,TRUE,FALSE,NULL)

To enable tracing for a session:
DBMS_MONITOR.SESSION_TRACE_ENABLE (SESSION_ID=>139, SERIAL_NUM=>53,
WAITS=>TRUE, BINDS=>FALSE);

To enable trace in the whole database
DBMS_MONITOR.DATABASE_TRACE_ENABLE

To enable trace in the instance level
DBMS_MONITOR.DATABASE_TRACE_ENABLE (INSTANCE_NAME=>'RAC1')

To disable tracing:
DBMS_MONITOR.CLIENT_ID_STAT_DISABLE(<Client_id>)
DBMS_MONITOR.SERV_MOD_ACT_TRACE_DISABLE('APPS1')
DBMS_MONITOR.DATABASE_TRACE_DISABLE(INSTANCE_NAME=>'RAC1')


/* Using the TRCSESS Tool to Analyze Trace Files */
show parameter USER_DUMP_DEST

trcsess output="hr_report.trc" service="APPS1" module="PAYROLL" action="bulk
load"

You can then run TKPROF against the consolidated trace file to generate a
report:
tkprof hr_report.trc output=hr_trc_report sys=np SORT=(EXEELA, PRSELA, FCHELA)
```

Enabling and Disabling Statistic Gathering for End to End Tracing

```
-- first make sure the trcing is enabled, see sections above.


-- Statistic Gathering for Client Identifier
select CLIENT_IDENTIFIER from V$SESSION where username='SA';

EXECUTE DBMS_MONITOR.CLIENT_ID_STAT_ENABLE(client_id => 'OE.OE');
EXECUTE DBMS_MONITOR.CLIENT_ID_STAT_DISABLE(client_id => 'OE.OE');


-- Statistic Gathering for Service, Module, and Action
EXECUTE DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(service_name => 'ACCTG',
module_name => 'PAYROLL');
EXECUTE DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(service_name => 'ACCTG',
module_name => 'GLEDGER', action_name => 'INSERT ITEM');
EXECUTE DBMS_MONITOR.SERV_MOD_ACT_STAT_DISABLE(service_name => 'ACCTG',
module_name => 'GLEDGER', action_name => 'INSERT ITEM');



/* Viewing the New Statistics */
```

```
DBA_ENABLED_AGGREGATIONS       enabled statistics aggregation
DBA_ENABLED_TRACES     enabled traces in the system
V$CLIENT_STATS  statistics on a client level (CLIENT_IDENTIFIER based)
V$SERVICE_STATS Displays basic performance statistics
V$SERV_MOD_ACT_STATS    statistics for a combination of serve /module/action
names.
```

Writing Efficient SQL

- Efficient WHERE Clauses
  - Select with high selectivity gets advantage from index. FTS performs better with low selectivity queries.
  - Whe using SQL functions, consider using function-based indexes.
- Using the Right Joins
  - Performing filtering operations early reduces the number of rows to be joined in later steps.
  - Always consider usinjg the equi join.
  - Join in the order that will produce the least number of rows as output to the parent step.
- Combine Multiples Scans with CASE Statements
  - When you need to calculate multiple aggregates from the same table, avoid writing a separate query for each aggregate. Use CASE statement instead.

```
-- the following selects:
SELECT COUNT (*)
 FROM employees
 WHERE salary < 2000;
SELECT COUNT (*)
 FROM employees
 WHERE salary BETWEEN 2000 AND 4000;
SELECT COUNT (*)
 FROM employees
 WHERE salary>4000;

-- can be re written as:
SELECT COUNT (CASE WHEN salary < 2000 THEN 1 ELSE null END) count1,
 COUNT (CASE WHEN salary BETWEEN 2001 AND 4000 THEN 1 ELSE null END) count2,
 COUNT (CASE WHEN salary > 4000 THEN 1 ELSE null END) count3
FROM employees;
```

- Efficient Subquery Execution
  - Oracle recommends using the IN clause if the subquery has the selective WHERE clause.
  - If the parent query contains the selective WHERE clause, use the EXISTS clause rather than the IN clause.

```
-- the follwoing  (selective predicate in the sub query)
WHERE EXISTS (SELECT 1 FROM orders o
 WHERE e.employee_id = o.sales_rep_id
 AND o.customer_id = 144);
-- can be converted into:
WHERE e.employee_id IN (SELECT o.sales_rep_id
 FROM orders o
 WHERE o.customer_id = 144);

-- in the following (selective predicate in the parent query):
WHERE e.department_id = 80
AND e.job_id = 'SA_REP'
AND e.employee_id IN (SELECT o.sales_rep_id FROM orders o);
-- can be rewritten as:
WHERE e.department_id = 80
```

```
AND e.job_id = 'SA_REP'
AND EXISTS (SELECT 1 FROM orders o WHERE e.employee_id = o.sales_rep_id)
```

- Avoid Transformed Columns in the WHERE Clause such as: charcol = numexpr, col1 = NVL (:b1,col1), NVL (col1,-999) = …., TO_DATE(), TO_NUMBER(), and so on.
- Add the predicate versus using NVL() technique.

```
SELECT employee_num, full_name Name, employee_id
FROM mtl_employees_current_view
WHERE (employee_num = NVL (:b1,employee_num)) AND (organization_id=:1)
ORDER BY employee_num;
```

- Write Separate SQL Statements for Specific Tasks: make a very complex statement slightly less complex by using the UNION ALL operator or use PL/SQL blocks, if possible.

```
-- this code won't use the index bcz somecol in both sides
..
WHERE ...
AND somecolumn BETWEEN DECODE(:loval, 'ALL', somecolumn, :loval)
AND DECODE(:hival, 'ALL', somecolumn, :hival);

-- it can be rewritten as
..
WHERE ...
AND somecolumn BETWEEN :loval AND :hival
AND (:hival != 'ALL' AND :loval != 'ALL')
UNION ALL
..
WHERE ...
AND (:hival = 'ALL' OR :loval = 'ALL');
```

- If possible, define the Join Order using ORDERED or STAR hints. Table with the lowest percentage of retireved compared to its tolal it candidate to be the driving table.

```
-- in this example, a is the driving table.
SELECT info
FROM taba a, tabb b, tabc c
WHERE a.acol BETWEEN 100 AND 200
AND b.bcol BETWEEN 10000 AND 20000
AND c.ccol BETWEEN 10000 AND 20000
AND a.key1 = b.key1
AND a.key2 = c.key2;

SELECT /*+ LEADING(e2 e1) USE_NL(e1) INDEX(e1 emp_emp_id_pk)
USE_MERGE(j) FULL(j) */
e1.first_name, e1.last_name, j.job_id, sum(e2.salary) total_sal
FROM employees e1, employees e2, job_history j
WHERE e1.employee_id = e2.manager_id
AND e1.employee_id = j.employee_id
AND e1.hire_date = j.start_date
GROUP BY e1.first_name, e1.last_name, j.job_id
ORDER BY total_sal;
```

- Modifying or Disabling Triggers and Constraints
- Restructuring the Data: like using vitual columns, adding new columns or using partitions.
- Maintaining Execution Plans Over Time: by using stored statistics or SQL plan baselines.
- Use DML with RETURNING Clause: INSERT, UPDATE, or DELETE... RETURNING modify and then return the data in one call.

```
var bnd1 NUMBER
var bnd2 VARCHAR2(30)
var bnd3 NUMBER

UPDATE employees
SET job_id ='SA_MAN', salary = salary + 1000,
department_id = 140
WHERE last_name = 'Jones'
RETURNING salary*0.25, last_name, department_id
INTO :bnd1, :bnd2, :bnd3;
```

- Consider using Test Case Builder.
- Consider using Bitmap Join Indexes on a star model query.
- Selecting the Best Join Order: If you're joining three tables, the one with the more restrictive filter (driving table) should be joined first (after the FROM keyword) to one of the other two tables.

```
-- this statement is less efficient than the following one
select /*+ ordered */
order_date, order_total, line_item_id id, product_name ,quantity,
quantity*unit_price item_tprice
from order_items i, PRODUCT_INFORMATION p, orders o
where
 order_date between to_date('01-01-2010','dd-mm-yyyy') and to_date('31-03-
2010','dd-mm-yyyy') and
 o.order_id=i.order_id and
 i.product_id=p.product_id;


-- more efficient
select /*+ ordered */
order_date, order_total, line_item_id id, product_name ,quantity,
quantity*unit_price item_tprice
from orders o, order_items i, PRODUCT_INFORMATION p
where
 order_date between to_date('01-01-2010','dd-mm-yyyy') and to_date('31-03-
2010','dd-mm-yyyy') and
 o.order_id=i.order_id and
 i.product_id=p.product_id;
```

Improving SQL Processing Techniques

Following are the tips to improve SQL processing, even if you cannot change the code:

- Using Partitioned Tables
- Using Compression Techniques

```
CREATE INDEX item_product_x
 ON order_items(product_id)
 TABLESPACE order_items_indx_01
 COMPRESS;
CREATE INDEX item_product_x
 ON order_items(product_id)
 TABLESPACE order_items_indx_01 COMPRESS;
```

- Using Materialized Views
- Using Stored Outlines to Stabilize the CBO (and SQL Plan Management in Oracle 11g).
- Using Parallel Execution

Using SQL Tuning Advisor

The advisor possible inputs are:

- a single statement
- SQL tuning set or STS
- a SQL identifier from the AWR
- a SQL identifier from the cursor cache
- To manage the Automatic SQL Tuning Advisor, see Using Automatic SQL Tuning Advisor.

```
-- required priv
grant advisor to youruser;


/* A single SQL Statement */
DECLARE
 my_task_name VARCHAR2(30);
 my_sqltext CLOB;
BEGIN
 my_sqltext := 'SELECT /*+ ORDERED */ *
  FROM employees e, locations l, departments d
  WHERE e.department_id = d.department_id AND
  l.location_id = d.location_id AND
 e.employee_id < :bnd';
my_task_name := DBMS_SQLTUNE.CREATE_TUNING_TASK(
 sql_text => my_sqltext,
 bind_list => sql_binds(anydata.ConvertNumber(90)),
 user_name => 'HR',
 scope => 'COMPREHENSIVE',
 time_limit => 60,
 task_name => 'my_sql_tuning_task',
 description => 'Task to tune a query on a specified employee');
END;
/


BEGIN
 DBMS_SQLTUNE.EXECUTE_TUNING_TASK( task_name => 'my_sql_tuning_task' );
END;
/


SET LONG 1000
SET LONGCHUNKSIZE 1000
SET LINESIZE 100
SELECT DBMS_SQLTUNE.REPORT_TUNING_TASK( 'my_sql_tuning_task') FROM DUAL;

-- you can query:
DBA_ADVISOR_TASKS
DBA_ADVISOR_FINDINGS
DBA_ADVISOR_RECOMMENDATIONS
DBA_ADVISOR_RATIONALE
DBA_SQLTUNE_STATISTICS
DBA_SQLTUNE_PLANS
```

Part 6 Oracle Automatic Storage Management (ASM)

Managing Oracle ASM

Obtaining Information about ASM Instance

```
-- asm: all diskgroups mounted    db: diskgroups mounted in asm and connected
to by db
select * from V$ASM_DISKGROUP;

-- asm: dbs connected to the asm     db: asms connected by db
select * from V$ASM_CLIENT;

-- asm: discovered disks    db: disks used by db
select * from V$ASM_DISK;

-- asm files
select * from V$ASM_FILE;

-- info about asm templates
select * from V$ASM_TEMPLATE;
```

Creating an ASM Instance

- The procedure below does not apply in 11g R2.

```
-- required services
<orahome>\bin\localconfig add

-- Creating the ASM Instance Manually (on Windows XP)
1. Building the ASM Candidate "disks": for testing or development purpose
 mkdir c:\asmdisks
 ASMTOOL -create c:\asmdisks\asmdiska1 150

-- you can use DBCA and avoid 2,3,4

2. Create a pfile with the name "init+ASM.ora" in the folder
<ORACLE_HOME>\database.
Insert the following parameters in the file:
INSTANCE_TYPE=ASM
_ASM_ALLOW_ONLY_RAW_DISKS = FALSE
DB_UNIQUE_NAME = +ASM
ASM_DISKSTRING ='C:\asmdisks\*'
LARGE_POOL_SIZE = 16M


3. Create the ASM instance service:
ORADIM -NEW -ASMSID +ASM -STARTMODE auto

4. Startup the instance
SET ORACLE_SID=+ASM
SQLPLUS / AS SYSDBA
SQL> STARTUP FORCE
SQL> SELECT PATH, MOUNT_STATUS FROM V$ASM_DISK;

SELECT instance_name FROM v$instance;
```

```
LSNRCTL STATUS


-- starting ASM in restricted mode: DB cannot access it (11g)
-- in the ASM instance level
SQL>STARTUP RESTRICT;
-- in the diskgroup level
SQL>ALTER DISKGROUP DATA MOUNT RESTRICTED;
-- check status of diskgroups
SQL>SELECT NAME,STATE FROM V$ASM_DISKGROUP;


-- creating a tablespace in an ASM diskgroup
CREATE TABLESPACE sample DATAFILE '+dgroup1';
```

Managing Disk Groups and Disks in ASM

Fundementals of Managing Disk Groups and Disks

```
/* Basics */
SELECT PATH, MOUNT_STATUS FROM V$ASM_DISK;

SQL> STARTUP NOMOUNT
SQL>
CREATE DISKGROUP dgroup1 NORMAL REDUNDANCY
 FAILGROUP controller1 DISK
  'c:\asmdisks\asmdiska1' name firsta size 150M,
  'c:\asmdisks\asmdiska2'
 FAILGROUP controller2 DISK
  'c:\asmdisks\asmdiskb1',
  'c:\asmdisks\asmdiskb2'
/

in the pfile:
ASM_DISKGROUPS=dgroup1

SQL> select NAME, SECTOR_SIZE, BLOCK_SIZE, ALLOCATION_UNIT_SIZE,
            STATE, TYPE, TOTAL_MB
     from V$ASM_DISKGROUP

restart

-- rebalancing speed
show parameter ASM_POWER_LIMIT
ALTER DISKGROUP dgroup1 REBALANCE POWER 5;


-- adding extra disks
ASMTOOL -create c:\asmdisks\asmdiska3 150
ASMTOOL -create c:\asmdisks\asmdiskb3 150

SQL>
ALTER DISKGROUP dgroup1 ADD
   FAILGROUP controller1 DISK 'c:\asmdisks\asmdiska3'
   FAILGROUP controller2 DISK 'c:\asmdisks\asmdiskb3'
/
```

```
ALTER DISKGROUP dgroup1 REBALANCE POWER 10
/



/* Dropping a disk */
ALTER DISKGROUP group1 DROP DISK disk5;
-- to cancel a pending drop:
ALTER DISKGROUP group1 UNDROP DISKS;

/* Dropping a Diskgroup */
in Oracle 10g:
# the group must be mounted on exactly one node
# non-empty dgroup
DROP DISKGROUP DATA INCLUDING CONTENTS;
# empty dgroup
DROP DISKGROUP DATA;

# If the diskgroup could not be mounted and the disks
#  were to be repurposed for non-ASM uses
use dd command in unix

# If the disks were to be repurposed as an ASM disk for a new diskgroup,
# then use the FORCE option in the CREATE DISKGROUP

in Oracle 11g:
# use FORCE if the group is not mounted (try mount it first)
DROP DISKGROUP DATA FORCE INCLUDING CONTENTS FORCE;


/* Checking Diskgroup */
-- 11g: you can validate the internal consistency of ASM diskgroup metadata.
Summary of errors is logged in the ASM alert log file.
-- check specific diskgroup with automatic repair
SQL>ALTER DISKGROUP data CHECK;
-- disable automatic repair
SQL>ALTER DISKGROUP data CHECK NOREPAIR;
SQL>ALTER DISKGROUP data CHECK REPAIR;
```

Managing Disk Groups Attributes

Oracle Database 11g introduces a new concept called ASM attributes at the diskgroup level. The attributes for the diskgroup can be established at create diskgroup time or can be modified using the `ALTER DISKGROUP` command later.

Following are the attributes you can set:

- Allocation unit (AU) sizes.
- The `compatible.rdbms` attribute.
- The `compatible.asm` attribute.
- `disk_repair_time` in units of minute (M) or hour (H) and is set by the `ALTER DISKGROUP` command
- The redundancy attribute for a specific template.
- The stripping attribute for a specific template.

All of the diskgroup attributes can be queried from the `V$ASM_ATTRIBUTE` view.

Consider the following examples:

```
CREATE DISKGROUP data
 disk '/dev/raw/raw1',
 ...
 attribute 'au_size' = '16M', 'compatible.asm' = '11.1';

ALTER DISKGROUP data SET ATTRIBUTE 'compatible.asm' = '11.1.0.0.0';

select NAME, VALUE from V$ASM_ATTRIBUTE where GROUP_NUMBER=1;
```

Variable AU Sizes

The default size of Allocation Unit (AU) is 1 MB which is sufficient for most regular databases. However, when you have databases with TB sizes, you will have enormous number of AUs. With Oracle 11g, AU size can be specified at diskgroup creation time to 1, 2, 4, 8, 16, 32, or 64MB in size. You can check the AU size through the following query:

```
select NAME, ALLOCATION_UNIT_SIZE from V$ASM_DISKGROUP;
```

Compatibility Settings

Compatibility in ASM is controlled in three ways, as shown below:

COMPATIBLE initialization parameter	The compatible initialization parameter can be set for either ASM or the database instance. It takes one of the following values: 10.1, 10.2, or 11.1. Setting the initialization parameter to a lesser value than the software release will exclude availability of the new features introduced in the new release.
RDBMS Compatibility	This is a diskgroup-level compatibility and is specified by setting the COMPATIBLE.RDBMS attribute. This attribute determines the minimum COMPATIBLE database initialization parameter setting for any database instance that uses the disk group. Its default value is 10.1.
ASM Compatibility	This is a diskgroup-level compatibility and is specified by setting the COMPATIBLE.ASM attribute. It determines the minimum software version for an ASM instance that uses the disk group.

If you assign any of the compatibility setting to a higher value, you cannot later reverse it to a lower value.

Following are some queries to obtain information about the compatibility settings:

```
-- diskgroup compatibility setting
select NAME, BLOCK_SIZE, ALLOCATION_UNIT_SIZE AU_SIZE, STATE,
COMPATIBILITY ASM_COMP, DATABASE_COMPATIBILITY DB_COMP
from V$ASM_DISKGROUP;

-- Compatibility of the database clients that use the ASM
select DB_NAME, STATUS,SOFTWARE_VERSION,COMPATIBLE_VERSION from V$ASM_CLIENT;
```

ASM Fast Mirror Resync

Any problems that make a failure group temporarily unavailable are considered *transient failures* that can be recovered by the ASM *fast mirror resync* feature. Disk path malfunctions; such as cable failures, host bus adapter failures, controller failures, or disk power supply interruptions; can cause transient failures.

ASM fast resync keeps track of pending changes to extents on an OFFLINE disk during an outage. The extents are resynced when the disk is brought back online.

Following are the steps to enable and handle this feature:

```
-- diskgroup compatibility must be set to 11.1
ALTER DISKGROUP dg1 SET ATTRIBUTE 'compatible.asm' = '11.1';
ALTER DISKGROUP dg1 SET ATTRIBUTE 'compatible.rdbms'='11.1';

-- specify the duration of the disk_repair_time (default is 3.6 hour)
ALTER DISKGROUP dg1 SET ATTRIBUTE 'disk_repair_time' = '5H'; -- in hours
ALTER DISKGROUP dg1 SET ATTRIBUTE 'disk_repair_time' = '40M'; -- minutes

-- verify the attribute settings
```

```
select NAME, VALUE from V$ASM_ATTRIBUTE;

-- if you get an offline disk because of a transient failure, you can see the
--  remaining time left in SECONDS before ASM drops an offline disk
select NAME, HEADER_STATUS, MOUNT_STATUS, MODE_STATUS, STATE, REPAIR_TIMER/60 from
V$ASM_DISK WHERE GROUP_NUMBER=1;

-- while the fix is in progress, if you want to reset the elapsed time, just take
--  the disk(s) offline
ALTER DISKGROUP dg1 OFFLINE DISK d3_0001;
ALTER DISKGROUP dg1 OFFLINE DISKS IN FAILGROUP f2;

-- you can also make a disk offline with a repair time different from its
--  disk_repair_time attribute
ALTER DISKGROUP dg1 OFFLINE DISK d3_0001 DROP AFTER 50m;

-- disks in a failure group (f2) can also be taken offline
ALTER DISKGROUP dg1 OFFLINE DISKS IN FAILGROUP f2 DROP AFTER 5m;

-- if the disk needs to be dropped immediately and before the repair time has expired
--  Note: ALTER DISKGROUP DROP DISK will not work
ALTER DISKGROUP dg1 OFFLINE DISK D3_0001 DROP AFTER 0m;

-- after the disk(s) are fixed, you can bring them online
ALTER DISKGROUP dg1 ONLINE ALL;
ALTER DISKGROUP dg1 ONLINE DISK d3_0001;
```

Monitoring Long-Running Operations

The `ALTER DISKGROUP DROP, RESIZE, and REBALANCE` commands return before the operation is complete. To monitor progress of these long-running operations, you can query the `V$ASM_OPERATION` fixed view.

`GROUP_NUMBER`	Disk group
`OPERATION`	Type of operation: `REBAL`
`STATE`	State of operation: `QUEUED` or `RUNNING`
`POWER`	Power requested for this operation
`ACTUAL`	Power allocated to this operation
`SOFAR`	Number of allocation units moved so far
`EST_WORK`	Estimated number of remaining allocation units
`EST_RATE`	Estimated number of allocation units moved per minute
`EST_MINUTES`	Estimated amount of time (in minutes) for operation termination

Migrating a Database to ASM

You can use OEM or RMAN as follows:

1. Shut down the database in a consistent mode by using the SHUTDOWN IMMEDIATE command.

2. Add the DB_CREATE_FILE_DEST and DB_CREATE_ONLINE_LOG_DEST_n parameters, as well as the new flash recovery area initialization parameters, DB_RECOVERY_FILE_DEST and DB_ RECOVERY_FILE_DEST_SIZE, to your database parameter file so you can use an OMF-based file system. Make sure that the two OMF parameters refer to the disk groups that you want to use in your ASM system.

DB_CREATE_FILE_DEST = '+dgroup1'

DB_RECOVERY_FILE_DEST = '+dgroup2'

3. Delete the control file parameter from the SPFILE, since Oracle will create new control files in the OMF file destinations by restoring them from the non-ASM database control files.

4. Start the database with the STARTUP NOMOUNT command:

RMAN> CONNECT TARGET;

RMAN> STARTUP NOMOUNT;

5. Restore the old control file in the new location, as shown here:

RMAN> RESTORE CONTROLFILE from '/u01/orcl/oradata/control1.ctl';

6. Mount the database:

RMAN> ALTER DATABASE MOUNT;

7. Use the following command to copy your database files into an ASM disk group:

RMAN> BACKUP AS COPY DATABASE FORMAT +dgroup1;

8. Use the SWITCH command to switch all datafiles into the ASM disk group:

RMAN> SWITCH DATABASE TO COPY;

At this point, all datafiles will be converted to the ASM type. You still have your original datafile copies on disk, which you can use to restore your database if necessary.

9. Open the database with the following command:

RMAN> ALTER DATABASE OPEN;

10. For each redo log member, use the following command to move it to the ASM system:

RMAN> SQL "alter database rename '/u01/test/log1' to '+dgroup1' ";

11. Archive the current online redo logs, and delete the old non-ASM redo logs. Since RMAN

doesn't migrate temp files, you must manually create a temporary tablespace using the

CREATE TEMPORARY TABLESPACE statement or:

ALTER TABLESPACE TEMP ADD TEMPFILE

ALTER DATABASE TEMPFILE '/u1/temp1' DROP

Moving a Tablespace to ASM

```
SQL>select name from v$datafile where ts# = (select x.ts# from v$tablespace x
where x.name='HRSTBS');

set oracle_sid=ora11g
```

```
rman target /

sql'alter tablespace hrstbs offline';

BACKUP AS COPY TABLESPACE hrstbs FORMAT '+dgroup1';

SWITCH TABLESPACE hrstbs TO COPY;

sql'alter tablespace hrstbs online';
```

Accessing an ASM instance from DB Console

```
/* Method 1 */
drop and recreate the em repository:
emca -deconfig dbcontrol db -repos drop
emca -config dbcontrol db -repos create

/* Method 2 */
- Edit your listener.ora and add an entry for the ASM instance in the SID_LIST

(SID_DESC =
(GLOBAL_DBNAME = My_DB_SID)
(ORACLE_HOME=d:\oracle\product\10.2.0\db_2)
(SID_NAME = +ASM)
)

- Restart your listener

- Access DBConsole (e.g. with SYS as SYSDBA)

- In the "Administration" or "Server" tab, click on the link called "Migrate
to ASM"

- Fill in the required fields and press continue (it won´t do anything yet.
It´s just the first step, to get ASM listen as a new OEM target).

- If it ever complains about ORA-1017, you need to recreate the ASM instance
paswordfile. Take a backup of it first if you feel better so.

The syntax for recreating the passwordfile can be, for example:

orapwd file=D:\oracle\product\10.2.0\db_2\database\PWD+ASM.ora
password=my_password force=y

(of course, you must replace the password and the path with the one that
serves your needs)

The ASM target should have been created.

Now, you only need to: click on the "Database" link on the right top of the DB
Console window, or access the DB Console page again. You should see the ASM
target link below the listener target. You only need to click the link and
you´ll get the ASM home administration page, from DBConsole.

If ASM hyperlink is not directly shown in the main DBconsole page, you can
still click on the Host name link in the left of the DBConsole main page (just
over the listener name link), and once there, again click in the "Targets"
```

```
link, and lastly in the "+ASM_your_DB_hostname" link, which then will take you
to the ASM Administration main page.

You also have the option of directly typing the following URL, after the ASM
target is created:
http://your_DB_hostname:your_DBConsole_port/em/console/database/osm/osmSitemap
?type=osm_instance&target=%2BASM_your_DB_hostname&event=doLoad
```

Managing ASM Files

Types of ASM Filenames

1. Fully Qualified ASM Filenames (System Alias)

You use this fully qualified name for *referencing* existing ASM files. Here's the syntax of an ASM file using a fully qualified filename:

```
+group/dbname/file_type/tag.file.incarnation
```

2. Numeric ASM Filenames

ASM derives *numeric filenames* from fully qualified ASM filenames and uses them to *refer* to existing files.

```
+group.file.incarnation
```

3. Alias ASM Filenames

You can use ASM alias files both when *creating* new ASM files and when *referring* to existing files. Alias ASM filenames mean that the files are not OMF-managed files. Thus, Oracle won't automatically remove these files when it does not have any further need for them.

```
+dgroup1/myfiles/control_file1
```

```
+dgroup2/mydir/second.dbf
```

4. Incomplete ASM Filenames

You can use an *incomplete ASM filename* only when *creating* files.

```
+dgroup1
```

```
+dgroup1(datafile)
```

Alias Filename Management

Creating Disk Group Directories for Alias Filenames

You must create a *directory structure* to support your alias filenaming conventions.

```
ALTER DISKGROUP dgroup1 ADD DIRECTORY '+dgroup1/mydir';
```

Using Templates with Aliases

```
dgroup(template_name)/alias
```

```
+dgroup1(spfile)/config1
```

Adding Aliases

You can add a filename alias or rename an existing alias name, using the `ADD ALIAS` or `RENAME ALIAS` clause of the `ALTER DISKGROUP` statement.

```
ALTER DISKGROUP dgroup1 ADD ALIAS '+dgroup1/mydir/second.dbf' FOR
'+dgroupA/sample/datafile/mytable.342.3'
```

You can retrieve created aliases using `v$ASM_ALIAS`. The `REFERENCE_INDEX` column is usable only for entries that are directory entries in the alias directory. For non-directory entries, it equals to zero.

Dropping Files and Aliases from a Disk Group

```
ALTER DISKGROUP dgroup1 DROP FILE '+dgroup1/payroll/compensation.dbf'
```

Using ASMCMD Utility

```
set oracle_home=c:\oracle\oradb11g
set oracle_sid=+ASM
asmcmd
help
```

Oracle 11g introduces new commands in the asmcmd utility and it also provides backward compatibility with Oracle Database 10g ASM instances. Following are summary of some of them:

Command Syntax	Description and Examples
lsct [-gH] [group]	Lists information about current ASM clients. >lsct dgroup1
lsdg [-gcH] [group]	lists all diskgroups and their attributes. >lsdg dgroup2
lsdsk [-ksptagcHI] [-d diskg_roup_name] [pattern]	lists the disks that are visible to ASM by scanning the disk headers of the disks seen by the value of the ASM_DISKSTRING >lsdsk -k -d DATA *_001 >lsdsk -s -d DATA *_001 >lsdsk -t -d DATA *_001 >lsdsk -c -t -d DATA *_001 >lsdsk -g -t -d DATA *_001
cp [-ifr] [connect_string:]src_fname [connect_string:]tgt_fname cp [-ifr] [connect_string:]src_fnameN, src_fnameN+1 ... [connect_string:]tgt_directory The connect_string is in the form of: user_name@host_name[.port_number].SID -i interactive -f force overwrite (aliases cannot be overwritten) -r recursive	Enables you to copy files between ASM disk groups on local instances to and from remote instances. >cp +dg1/vdb.ctf1 /backups/vdb.ctf1 >cp /home/oracle/encrypted.dmp +dg1 >cp vdb.ctf1 /tmp # the target ASM instance must be registered with the LISTENER >cp +DATA/DBA11g/DATAFILE/DOCS_D1.289.631914611 sys@rac1.+ASM:+DATA/DBA11g1/datafile/xxx
remap	Repairs a range of physical blocks on disk (only blocks exhibiting read disk I/O errors are repaired) excluding those with corrupted contents. Internally, it reads the blocks from a good copy of an ASM mirror and rewrites them to an alternate location on disk, if the blocks on the original location cannot be properly read. remap <disk group name> <disk name> <block range> > remap DISK_GRP1 DATA_0001 5000-5999

Backing up and Restoring Diskgroup Metadata

The md_backup command captures information about ASM disks, diskgroup and failure group configurations, and template and alias directory structures, and stores them in a user-designated backup text file. Following is the basic syntax of the command:

```
md_backup [-b <backup_file_path> ] [-g diskgroup_name [-g diskgroup_name …]]
```

Following is an example of using the command:

```
md_backup –b /tmp/asm_backup.mdb -g dg1 –g dg2
```

If the backup file already exists, you should remove it before issuing the command.

If you issue the md_backup command without any option, it creates a file named as ambr_backup_intermediate_file which contains the metadata information of all the mounted diskgroups.

The md_restore command reads the backup file and restores a disk group. You can set its options to build a script file that contains the SQL statements required to rebuild the ASM components from the backup file. Following is the syntax of the command and description of its switches:

```
md_restore -b <backup_file> [-li]
 [-t (full)|nodg|newdg] [-f <sql_script_file>]
 [-g '<diskgroup_name>,<diskgroup_name>,...']
 [-o '<old_diskgroup_name>:<new_diskgroup_name>,...']
```

`-t` type of restore.

`full` tag specifies that all the diskgroups should be re-created using the same configuration from the MDB backup file.

nodg restore metadata only and skip the diskgroup creation.

newdg create disk group with a different name and restore metadata; `-o` is required. This tag is used to allow the user to change diskgroup name.

`-f` write SQL commands to `<sql_script_file>` instead of executing them.

`-o` override option is used only with the newdg option to remap the diskgroup name, disk name, paths, and failure groups.

`-i` ignore errors. By default, the command aborts when it encounters an error.

`-l` log all messages to a log file.

Following are examples of using the command:

```
/* General Examples */
# To perform a restore of the dg1 diskgroup from the MDB backup file, use this:
md_restore –b /tmp/backupfile –t full –g dg1 -i

# To just restore the metadata for the dg1 diskgroup (the diskgroup already
exists).
md_restore –b /tmp/backupfile –t nodg –g dg1 –i

# To create a different diskgroup name:
md_restore –b /tmp/backupfile –t newdg -o "DGNAME=dg1:dg3" –i

# To apply the override options as specified in the dg_over.txt file and restore
#  from the backup file:
md_restore –b /tmp/backupfile –t newdg –of /tmp/dg_override.txt –i


/* Scenario Example */
1. Back up a tablespace exising in a disk group:
RMAN> BACKUP TABLESPACE users;
2. Create a directory named test in the disk group DGROUPA. Also create an alias
called
+DGROUPA/test/users.f that points to the ASM datafile that contains the users
tablespace:
ASMCMD> mkdir +DGROUPA/test
ASMCMD> mkalias TBSSRA.123.123456789 +DGROUPA/test/users.f
3. Back up the metadata for the disk group DGROUPA using the md_backup command:
ASMCMD> md_backup –g dgroupA
The md_backup command stores the backup metadata in the text file named
ambr_backup_
intermediate in the current directory.
4. Simulate a disk failure by dropping the disk group DGROUPA:
SQL> ALTER DISKGROUP dgroup1 DISMOUNT FORCE;
SQL> DROP DIKSGROUP dgroup1 FORCE INCLUDING CONTENTS;
```

```
The DISMOUNT FORCE clause in the ALTER DISKGROUP command dismounts the disk group
and
force drops it.
5. Execute the md_restore command to restore the ASM metadata for the dropped disk
group:
ASMCMD> md_restore –b ambr_backup_intermediate_file
–t full –g data
6. Using the backup of the users tablespace from step 1, restore the users
tablespace:
RMAN>RESTORE TABLESAPCE users;
7. Exit from RMAN once the restore is completed.
```

Note that md_backup is a backup of the metadata of the ASM instance. The data is being backed up by RMAN. After the diskgroup is created, along with all the directories, you can restore the RMAN backup to the diskgroup.

Bad Block Recovery

If ASM cannot read a physical block from a disk, it considers that the block has IO error. In this case, ASM will automatically read a mirrored block and write a relocated copy to produce successful copy. However, you can manually repair blocks that have read disk I/O errors using the remap command. Following is the syntax of the command:

```
remap <diskgroup name> <disk name> <block range>
```

Using SYSASM Privilege and OSASM Group

This feature introduces a new SYSASM privilege that is specifically intended for performing ASM administration tasks. Using the SYSASM privilege instead of the SYSDBA privilege provides a clearer division of responsibility between ASM administration and database administration.

Following are code examples illustrating how to use this privilege:

```
-- grant the privilege
GRANT SYSASM TO firas;

-- check the granted privilege
SELECT * FROM V$PWFILE_USERS;

-- ASM management commands are available to Adam
CONNECT firas/his_password
ALTER DISKGROUP dg1 DISMOUNT;
ALTER DISKGROUP dg2 MOUNT;
.. and so on.
```

Be aware that users with SYSOPER privilege have some ASM privileges. Following table shows available and restricted ASM privilege for users with SYSOPER privilege:

Avaiable ASM Privilege	*Restricted ASM Privilege*
STARTUP AND SHUTDOWN ALTER DISKGROUP MOUNT ALTER DISKGROUP DISMOUNT ALTER DISKGROUP ONLINE DISK ALTER DISKGROUP OFFLINE DISK ALTER DISKGROUP REBALANCE ALTER DISKGROUP CHECK	CREATE DISKGROUP / DISK DROP DISKGROUPS / DISKS ALTER DISKGROUP / DISK RESIZE

OSASM is a new operating system group that is used exclusively for ASM. Members of the OSASM group can connect as SYSASM using operating system authentication and have full access to ASM.

Manually Upgrading Oracle AS from 10g to 11g

Following are the steps you follow to upgrade an existing Oracle 10g ASM to 11g:

1. Install the Oracle Database 11g software to a new ORACLE_HOME directory.
2. Update the `/etc/oratab` or `/var/opt/oracle/oratab` file with the new ASM ORACLE_HOME location.
3. Copy the ASM initialization file from the old ORACLE_HOME to the new one.
4. Edit any directory-based parameters (such as `diag` and `dump`) in the ASM initialization file as required.
5. If you are upgrading a non-RAC ASM instance, you should reconfigure the Oracle CSS using the new ORACLE_HOME. You can do this by executing the `localconfig` command from the new home. Once the CSS configuration is complete, you need to change your ORACLE_HOME to the new Oracle version 11.1 ORACLE_HOME and start the ASM instance.

```
cd $ORACLE_HOME/bin
 # ./localconfig reset
```

6. If you are upgrading a ASM instance in a RAC environments, you can modify the new ASM home within the OCR using the `srvctl` utility as follows:

```
srvctl modify asm -n racnode1 -i +ASM1 -o /apps/oracle/product/11.1.0/asm -p
init+ASM1.ora
```

7. Grant the `SYSASM` role to the SYS

```
GRANT SYSASM to sys;
```

8. If you have obsolete initialization parameters, you can address them now. To get a listing of all the obsolete initialization parameters, refer to the ASM alert log file.

Verifying Manually ASM Device

```
# To verify asmlib status:
## Both should be [OK]
/etc/init.d/oracleasm status
Checking if ASM is loaded:                                  [  OK  ]
Checking if /dev/oracleasm is mounted:                [  OK  ]


# To verify user setting in asmlib:
uid=1001(grid) gid=1000(oinstall) groups=1000(oinstall)

/usr/sbin/oracleasm configure
ORACLEASM_ENABLED=true
ORACLEASM_UID=grid
ORACLEASM_GID=oinstall
ORACLEASM_SCANBOOT=true
ORACLEASM_SCANORDER=""
ORACLEASM_SCANEXCLUDE=""


# To verify disk
/etc/init.d/oracleasm listdisks
DISKCLU
ls -l /dev/oracleasm/disks
..
brw-rw----    1 grid oinstall        8,  33 Sep 16 09:41 DISKCLU
```

```
# Disk DISKCLU is available and readable from above output.
dd if=/dev/oracleasm/disks/DISKCLU of=/dev/null bs=1024k count=1
1+0 records in
1+0 records out
```

Part 7 Oracle Real Application Cluster

Oracle RAC Possible Installation Configurations

For Oracle Software homes, Voting and OCR, db files:

- NTFS_RAW_ASM
- NTFS_RAW_RAW
- NTFS_OCFS_OCFS
- OCFS_OCFS_OCFS

Installing Oracle 10g R2 RAC on Enterprise Linux 4

Note: The metalink document RAC Starter Kit and Best Practices (Linux) [ID 811306.1] is a good source reference for this task.

Installation Environment

- Emulation software: VMWare Workstation ACE Edition 6.0.5 or VMWare Server 2.
- RAC Nodes: 2 nodes with 2 GB RAM each , 2 ethernet cards.
- OS: Oracle Linux Enterprise 4.5 for x86: kernel 2.6.9

Required Software

- Oracle 10g R2 Clusterware for Linux x86 32-bit
- Oracle Database 10g Release 2 for Linux x86 32-bit

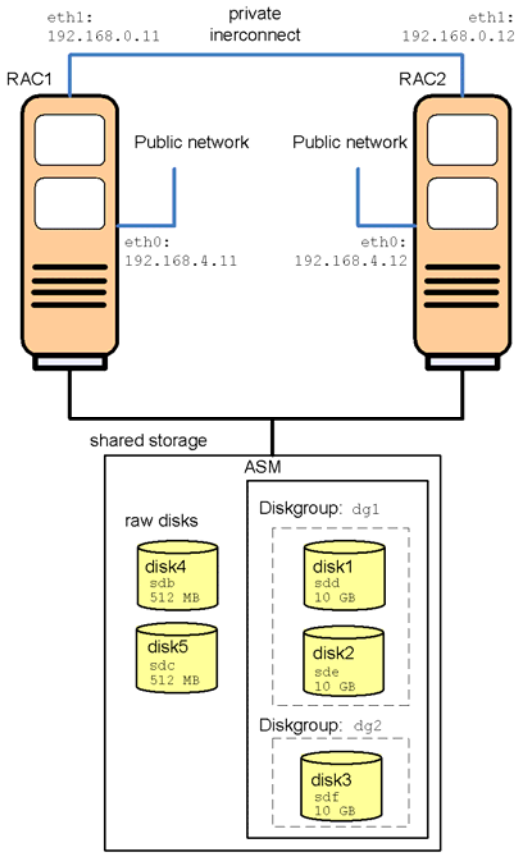
Used Hardware

- In the VMWare: create one virtual machine (rac1) with the following specs:
  - 2 GB RAM
  - Two ethernet cards: both can be configured as bridged or host-only in VMware.
  - One local hardisk with 20 GB
  - CPU Count: 2
  - Create a folder in the same directory structure level as the parent folder containing the created virtual machine. Give it a meaningful name like 'shared_disks'. Create in that folder three disks of 10 GB and two of 512 MB. All of them are of LSI Logic type and SCISI Persistent. Make sure they are on SCISI controller different from the SCSI controller of the local hardisk. For example, if the SCSI controller of local hardisk is SCISI0, make those disks on controller SCSI1.

Installation Plan

1. Preinstallation tasks:
   - Hardware requirements
   - Software requirements
   - Environment configuration
2. Oracle Clusterware installation
3. Oracle Database 10g Software Installation
4. Apply Patchset 3 (10.2.0.4) for Clusterware and Database Software
5. Install EM Agent in cluster nodes (if required)
6. Configure Listeners
7. Perform ASM installation

8. Perform cluster database creation
9. Complete postinstallation tasks
10. Useful postinstallation tasks



1. Preinstallation tasks

- Install Oracle Enterprise Linux in the first local hardisk. Install nothing in the remaining disks.

  Note: for a production system, consider becoming an Oracle Unbreakable Linux customer and register your server on the Unbreakable Linux Network.

  - Give the first ethernet card IP 192.168.4.11 and the second 192.168.0.11 and the hostname `rac1.mydomain.com`. Define a gateway. If it does not exist, make it same as the host IP address.
  - Insall the following packages:
    - Desktop Environments
      - GNOME Desktop Environment
    - Desktop
      - X Window System
      - Gnome

        - Applications
            - Graphical Internet (optional)
        - Servers
            - Do not select anything in this group.
        - Development
            - Development Tools
        - System
            - Administration Tools
            - System Tools
                - Add the package 'sysstat' by clicking on the Details link and selecting "sysstat - The sar an iostat system monitoring commands." from the Optional Packages list.
        - Miscellaneous
            - Do not select anything in this group.
- Complete the installation
- Install further packages:

```
# to know distribution and version of Linux
cat /etc/issue
# to know kernel version (and its errata level)
uname -r

# from CD 3
rpm -Uvh libaio*
rpm -Uvh openmotif21-2.1.30-11.RHEL4.6.i386.rpm
rpm -Uvh openmotif-2.2.3-10.1.el4.i386.rpm

# those packages downloaded from http://rpm.pbone.net
rpm -e compat-libstdc++-296-2.96-132.7.2
rpm -Uvh compat-libstdc++-7.3-2.96.128.i386.rpm
rpm -Uvh compat-libstdc++-devel-7.3-2.96.128.i386.rpm
rpm -Uvh compat-gcc-7.3-2.96.128.i386.rpm
rpm -Uvh compat-gcc-c++-7.3-2.96.128.i386.rpm

# confirm the required packages are installed:
rpm -qa|grep gcc-
rpm -qa|grep glibc-
rpm -qa|grep compat-db-
rpm -qa|grep compat-gcc-
rpm -qa|grep compat-gcc-c++-
rpm -qa|grep compat-libstdc++-
rpm -qa|grep compat-libstdc++-devel-
rpm -qa|grep control-center-2.8.0
rpm -qa|grep openmotif21-
rpm -qa|grep setarch-

# SELINUX must be disabled
cat /etc/selinux/config | grep SELINUX=
vi /etc/selinux/config
SELINUX=disabled
shutdown -h now -r
```

```
# Install ASMLib 2.0 packages
# install the library for your kernel and CPU type
# oracleasm-*-version.cpu_type.rpm
# check installed packages
rpm -qa|grep asm
# install the packages from CD3
rpm -Uhv oracleasm-support-2.0.3-2.i386.rpm
rpm -Uhv oracleasm-2.6.9-55.0.0.0.2.EL-2.0.3-2.i686.rpm
rpm -Uhv oracleasm-2.6.9-55.0.0.0.2.ELsmp-2.0.3-2.i686.rpm

# download the package Userspace Library
# from
http://otn.oracle.com/software/tech/linux/asmlib/files/RPMS/rhel4/x86/2.0.4/or
acleasmlib-2.0.4-1.el4.i386.rpm
rpm -Uvh oracleasmlib-2.0.4-1.el4.i386.rpm
```

- Check the hardware requirements

```
# Hardware Requirements (in cluster nodes)
# At least 1 GB of physical memory
grep MemTotal /proc/meminfo
# swap space: twice the amount of physical memory
grep SwapTotal /proc/meminfo
# if you don't have enought swap,
#  you can add swap space by creating a temporary swap file.
#  let's say about 500MB:
dd if=/dev/zero of=tempswap bs=1k count=500000
chmod 600 tempswap
mke2fs tempswap
mkswap tempswap
swapon tempswap

# 400 MB disk space in /tmp
df -k /tmp
# 4 GB of disk space for Oracle software
df

The size of the shared memory should be at least the greater of
MEMORY_MAX_TARGET and MEMORY_TARGET for each Oracle instance on the computer.
To determine the amount of shared memory available, enter the following
command:
# df -h /dev/shm/
```

- Create the required network configuration (rac2 will be created later):

```
# Network names Resolution
# configure /etc/hosts if no domain server is used (both nodes)
vi /etc/hosts
127.0.0.1       localhost.localdomain   localhost
#Public
192.168.4.11    rac1.mydomain.com    rac1
192.168.4.12    rac2.mydomain.com    rac2
#VIP
192.168.4.13    rac1-vip.mydomain.com   rac1-vip
192.168.4.14    rac2-vip.mydomain.com   rac2-vip
#Inter-connect
192.168.0.11    rac1-priv.mydomain.com       rac1-priv
192.168.0.12    rac2-priv.mydomain.com       rac2-priv
```

Note: To prevent network hangs with failovers from public to virtual IP addresses with RAC databases using NAS devices or NFS mounts, enter the following command as root to enable the Name Service Cache Daemon: `/sbin/service nscd start`

- Create and configure the required OS users and groups

  Note: userid and groupid must be the same in all nodes. You can check them by `id oracle` command.

```
# all group and user ids on all the nodes must have identical id
# inventory and OSDBA groups (if needed, use -g <number> to specify the id)
# inventory group
groupadd -g 501 oinstall
groupadd -g 502 dba
# oracle software owner user (take note of userid)
/usr/sbin/useradd -u 200 -g oinstall -G dba oracle
passwd oracle
# make sure nobody user exists (if not there, create it useradd nobody)
id nobody

# The oracle User Environment
# in /home/oracle/.bash_profile
# export DISPLAY if required
export ORACLE_BASE=/u01/app/oracle
if [ $USER = "oracle" ]; then
 if [ $SHELL = "/bin/ksh" ]; then
  ulimit -p 16384
  ulimit -n 65536
 else
  ulimit -u 16384 -n 65536
 fi
 umask 022
fi
export EDITOR=vi
export ORACLE_HOME=$ORACLE_BASE/product/10.2.0/db_1
export ORA_CRS_HOME=/u01/crs
export ORACLE_PATH=$ORACLE_BASE/common/oracle/sql:.:$ORACLE_HOME/rdbms/admin
export ORACLE_SID=rac1
export NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
export NLS_DATE_FORMAT="mm/dd/yyyy hh24:mi:ss"
export PATH=.:${PATH}:$HOME/bin:$ORACLE_HOME/bin:$ORA_CRS_HOME/bin
export PATH=${PATH}:/usr/bin:/bin:/usr/bin/X11:/usr/local/bin
export PATH=${PATH}:$ORACLE_BASE/common/oracle/bin
export ORACLE_TERM=xterm
export TNS_ADMIN=$ORACLE_HOME/network/admin
export ORA_NLS10=$ORACLE_HOME/nls/data
export TNS_ADMIN=$ORACLE_HOME/network/admin
export ORA_NLS10=$ORACLE_HOME/nls/data
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$ORACLE_HOME/oracm/lib
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib:/usr/lib:/usr/local/lib
export CLASSPATH=$ORACLE_HOME/JRE
export CLASSPATH=${CLASSPATH}:$ORACLE_HOME/jlib
export CLASSPATH=${CLASSPATH}:$ORACLE_HOME/rdbms/jlib
export CLASSPATH=${CLASSPATH}:$ORACLE_HOME/network/jlib
export THREADS_FLAG=native
# should NOT be on a shared disk
export TEMP=/tmp
export TMPDIR=/tmp
```

- Configure kernel parameters and shell limits

Note: If you make a mistake with a parameter setting and your system does not start, then you must start Linux in the single-user runlevel (runlevel 1). At this runlevel, the /etc/sysctl.conf file is not run.

```
# Configuring kernel parameters and shell limits

# User Shell Limits
# memlock is used to increase the per-process max locked memory
vi /etc/security/limits.conf
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft memlock 3145728
oracle hard memlock 3145728

vi /etc/pam.d/login
session required /lib/security/pam_limits.so

# they can be tuned for a production db
# Append the following to the /etc/sysctl.conf file as the root user:
vi /etc/sysctl.conf
kernel.sem = 250 32000 100 128
# It should be equal to or larger than the largest SGA (max 4 GB)
kernel.shmmax = 536870912
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default = 4194304
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 262144
# to take immediate effect
/sbin/sysctl -p
```

- Configure hangcheck-timer kernel module:

```
# check hangcheck-timer Module Configuration
# with this module, if the kernel hangs, the machine will reboot
# verify the module is loaded
/sbin/lsmod | grep -i hang

# if not loaded, load it
vi /etc/modprobe.conf
options hangcheck-timer hangcheck_tick=30 hangcheck_margin=180
# execute and add in the file
vi /etc/rc.local
/sbin/modprobe hangcheck-timer
```

- Partition the disks and prepare the raw disks

```
# Partition the devices
# for the disks /dev/sdb .. /dev/sdf
fdisk /dev/sdb
# answers: "n", "p", "1", "Return", "Return", "p" and "w"
Note: if the following message appears after the "w" command:
 WARNING: Re-reading the partition table failed with error 16: Device or
 resource  busy.
then, you can avoid restarting the machine by the following command:
partprobe
```

```
# to make sure partions are created
ls -lX /dev/sd*
brw-r----- 1 root disk 8,  0 Dec  1 20:35 /dev/sda
brw-r----- 1 root disk 8,  1 Dec  1 20:36 /dev/sda1
brw-r----- 1 root disk 8,  2 Dec  1 20:35 /dev/sda2
brw-r----- 1 root disk 8, 16 Dec  1 21:55 /dev/sdb
brw-r----- 1 root disk 8, 17 Dec  1 21:55 /dev/sdb1
brw-r----- 1 root disk 8, 32 Dec  1 21:57 /dev/sdc
brw-r----- 1 root disk 8, 33 Dec  1 21:57 /dev/sdc1
brw-r----- 1 root disk 8, 48 Dec  1 21:57 /dev/sdd
brw-r----- 1 root disk 8, 49 Dec  1 21:57 /dev/sdd1
brw-r----- 1 root disk 8, 64 Dec  1 21:58 /dev/sde
brw-r----- 1 root disk 8, 65 Dec  1 21:58 /dev/sde1
brw-r----- 1 root disk 8, 80 Dec  1 21:58 /dev/sdf
brw-r----- 1 root disk 8, 81 Dec  1 21:58 /dev/sdf1

# binding sdisks to raw devices
raw /dev/raw/raw1 /dev/sdb1
raw /dev/raw/raw2 /dev/sdc1
# list the raw devices
raw -qa
# add to the file
vi /etc/sysconfig/rawdevices
/dev/raw/raw1 /dev/sdb1
/dev/raw/raw2 /dev/sdc1

# Adjust the raw devices permission settings:
# Run the following commands AND add them the /etc/rc.local file:
# will be used by OCR
chown root:oinstall /dev/raw/raw1
chmod 660 /dev/raw/raw1
# will be used by voting disk
chown oracle:oinstall /dev/raw/raw2
chmod 644 /dev/raw/raw2
# will be used by ASM, if it is using raw device (NOT IN THIS CONFIGURATION)
# chown oracle:oinstall /dev/raw/raw3
# chmod 660 /dev/raw/raw3

# start the service
service rawdevices restart
```

- Create the required directories for the Oracle clusterware and database software

```
# to know if there is an existing oracle inventory
# from its output, ORACLE_BASE will be parent of oraInventory
more /etc/oraInst.loc
# to identify existing Oracle home directories
more /etc/oratab

# in the example above, /u01 should be owned by the root user
#  and writable by group oinstall
# directories must be same in all nodes:
mkdir -p /u01/app/oracle/product/10.2.0/db_1
mkdir /u01/stage10g
# clusterware must NOT be subdirectory of the ORACLE_BASE
mkdir /u01/crs
chown -R oracle:oinstall /u01/app/oracle
```

```
chmod -R 775 /u01/app/oracle
chown oracle:oinstall /u01/stage10g
chown -R oracle:oinstall /u01/crs
chmod -R 775 /u01/crs
```

- Disable screensavers on host & guest machines.
  - In Oracle Linux: Applications-> Preferences-> Screen Saver-> Mode: Disable Screen Saver
  - Do the same after logging off and logging on again as oracle user.
- Shutdown rac1
- Edit the VMware file (with vmx extensions) and add the following entry to allow sharing the disks (make sure the scsi controller number is the one you used):

```
disk.locking = "FALSE"
diskLib.dataCacheMaxSize = "0"
diskLib.dataCacheMaxReadAheadSize = "0"
diskLib.dataCacheMinReadAheadSize = "0"
diskLib.dataCachePageSize = "4096"

scsi1.sharedBus = "virtual"
scsi1:0.deviceType = "disk"
scsi1:1.deviceType = "disk"
scsi1:2.deviceType = "disk"
scsi1:3.deviceType = "disk"
scsi1:4.deviceType = "disk"
scsi1:5.deviceType = "disk"
```

- Copy the folder containing rac1 into a new folder in the same directory structure level. Let's name it "rac2". This will be the second node in the cluster.
- Edit the VMware file of rac1 and edit the following: displayName = "rac2"
- Open rac2 then probe its network cards, change it's IP addresses: IP 192.168.4.12 and 192.168.0.12. It's hostname to rac2.mydomain.com. You can use system-config-network-gui. Activate the network and reboot the vm.
- In rac2, perform:

```
# change the variable in the file
vi /home/oracle/.bash_profile
export ORACLE_SID=rac2
```

- Start rac1
- Configure ASM drivers:

```
# as root  ( on ALL NODES )
service oracleasm configure

Default user to own the driver interface []: oracle
Default group to own the driver interface []: dba
Start Oracle ASM library driver on boot (y/n) [n]: y
Fix permissions of Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration:           [  OK  ]
Creating /dev/oracleasm mount point:                       [  OK  ]
Loading module "oracleasm":                                [  OK  ]
Mounting ASMlib driver filesystem:                         [  OK  ]
Scanning system for ASM disks:                             [  OK  ]

# As the root user on node node1 (ONLY ONE NODE)
service oracleasm createdisk DISK1 /dev/sdd1
```

```
service oracleasm createdisk DISK2 /dev/sde1
service oracleasm createdisk DISK3 /dev/sdf1
# if any of the commands above fails, you can delete the disk
service oracleasm deletedisk DISK1
# in ALL NODES
service oracleasm scandisks
service oracleasm listdisks
```

- Configure SSH in all the nodes:

Note: scp and ssh must be located in the path /usr/local/bin. If not, then create a symbolic link in /usr/local/bin to the location where scp and ssh are found.

```
# Configuring SSH for Remote Installation
# make sure the ssh is running
# the following command shoudl return ssh process id
pgrep sshd

vi /etc/hosts.equiv
# add node names to /etc/hosts.equiv (all nodes)
rac1
rac2
# as oracle in node1
#  whenever prompted for a passphrase leave it empty and press return
mkdir ~/.ssh
chmod 700 ~/.ssh
/usr/bin/ssh-keygen -t rsa
# as oracle in node2
mkdir ~/.ssh
chmod 700 ~/.ssh
/usr/bin/ssh-keygen -t rsa
# as oracle on node1
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
ssh rac2 cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
scp ~/.ssh/authorized_keys rac2:/home/oracle/.ssh/

# Perform the following as the oracle user on node1 and then node2
#  copy paste the following command in both nodes twice:
ssh rac1 date;ssh rac2 date;ssh rac1.mydomain.com date;ssh rac2.mydomain.com
date;ssh rac1-priv date;ssh rac2-priv date;ssh rac1-priv.mydomain.com date;ssh
rac2-priv.mydomain.com date;ssh localhost.localdomain date;ssh localhost date
# Note: the keys are session dependent. it's gone from memory on log off
# Execute (to load the keys in the memory):
exec /usr/bin/ssh-agent $SHELL
/usr/bin/ssh-add
```

2. Oracle Clusterware installation

```
# make sure system time of rac2 should be behind the system time of rac1
date; ssh rac2 date

# Identify the shared disks to use
#  /dev/sdb & /dev/sdc will be used for OCR & Voting by the Cluster software
/sbin/fdisk –l

# unzip the Oracle 10g R2 Clusterware for Linux in the stage folder (rac1)

# If you plan to use Oracle Clusterware on x86 (64-bit) (but not on
#  Linux Itanium 64-bit), then you must run the rootpre.sh on all nodes
```

```
su -
cd /u01/stage10g/clusterware/rootpre
./rootpre.sh

# install cluvfy (in all nodes)
# in node1
su -
cd /u01/stage10g/clusterware/rpm
rpm -q cvuqdisk-1.0.1
export CVUQDISK_GRP=dba
rpm -ivh cvuqdisk-1.0.1-1.rpm
# in node2
su -
export CVUQDISK_GRP=dba
rpm -ivh cvuqdisk-1.0.1-1.rpm

# Verify Cluster Setup with cluvfy
# as oracle
# if the user equivalency is not loaded
exec /usr/bin/ssh-agent $SHELL
/usr/bin/ssh-add
#  then proceed:
cd /u01/stage10g/clusterware/cluvfy
export CV_NODE_ALL=rac1,rac2
./runcluvfy.sh stage -pre crsinst -n rac1,rac2 -verbose
# If only the VIP was the problem, then you're going in the right way!


# lunch OUI from the clusterware ( as oracle from node1)
# if logged in in Genome using another user, log out and log in as oracle
# to reload the user equivalency, if logged off: (as oracle)
exec /usr/bin/ssh-agent $SHELL
/usr/bin/ssh-add
# if not turned off, turn off the screensaver IN ALL NODES
# then process:
cd /u01/stage10g/clusterware
./runInstaller


>Inventory Directory (displayed only in first time)
 /u01/app/oracle/oraInventory
 oinstall

>Home Details
/u01/crs

>Product-Specific Prerequisite Checks
 They all should succeed

>Specify Cluster Configuration
Select 'Add' and add the details for node rac2. Enter all details. 'OK' to
continue.
rac2.mydomain.com
rac2-priv.mydomain.com
rac2-vip.mydomain.com
Click 'Next'
```

```
>Specify Network Interface Usage
Change the eth0 Interface Type to Public.
'Ok' to continue

>Specify Oracle Cluster Registry (OCR) Location
Select External Redundancy and
specify the first raw device /dev/raw/raw1 as the OCR location
'Next' to continue.

>Specify Voting Disk Location
Select External Redundancy and
specify the first raw device /dev/raw/raw2 as the voting disk location
'Next' to continue.

>Summary
Review the summary and select 'Install' to start the installation

>Configuration Scripts
as the root user on each node, run the scripts. Do not run the scripts
simultaneously on both nodes.
if there is an error in scripts execution, it's reported in /u01/
/crs/log/<hostname>
```

output of running the script in node1:

```
Checking to see if Oracle CRS stack is already configured
/etc/oracle does not exist. Creating it now.

Setting the permissions on OCR backup directory
Setting up NS directories
Oracle Cluster Registry configuration upgraded successfully
assigning default hostname rac1 for node 1.
assigning default hostname rac2 for node 2.
Successfully accumulated necessary OCR keys.
Using ports: CSS=49895 CRS=49896 EVMC=49898 and EVMR=49897.
node <nodenumber>: <nodename> <private interconnect name> <hostname>
node 1: rac1 rac1-priv rac1
node 2: rac2 rac2-priv rac2
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
Now formatting voting device: /dev/raw/raw2
Format of 1 voting devices complete.
Startup will be queued to init within 90 seconds.
Adding daemons to inittab
Expecting the CRS daemons to be up within 600 seconds.
CSS is active on these nodes.
        rac1
CSS is inactive on these nodes.
        rac2
Local node checking complete.
Run root.sh on remaining nodes to start CRS daemons.
```

output of running the script in node2:

```
Checking to see if Oracle CRS stack is already configured
/etc/oracle does not exist. Creating it now.

Setting the permissions on OCR backup directory
Setting up NS directories
```

```
Oracle Cluster Registry configuration upgraded successfully
clscfg: EXISTING configuration version 3 detected.
clscfg: version 3 is 10G Release 2.
assigning default hostname rac1 for node 1.
assigning default hostname rac2 for node 2.
Successfully accumulated necessary OCR keys.
Using ports: CSS=49895 CRS=49896 EVMC=49898 and EVMR=49897.
node <nodenumber>: <nodename> <private interconnect name> <hostname>
node 1: rac1 rac1-priv rac1
node 2: rac2 rac2-priv rac2
clscfg: Arguments check out successfully.

NO KEYS WERE WRITTEN. Supply -force parameter to override.
-force is destructive and will destroy any previous cluster
configuration.
Oracle Cluster Registry for cluster has already been initialized
Startup will be queued to init within 90 seconds.
Adding daemons to inittab
Expecting the CRS daemons to be up within 600 seconds.
CSS is active on these nodes.
        rac1
        rac2
CSS is active on all nodes.
Waiting for the Oracle CRSD and EVMD to start
Waiting for the Oracle CRSD and EVMD to start
Waiting for the Oracle CRSD and EVMD to start
Waiting for the Oracle CRSD and EVMD to start
Waiting for the Oracle CRSD and EVMD to start
Waiting for the Oracle CRSD and EVMD to start
Waiting for the Oracle CRSD and EVMD to start
Oracle CRS stack installed and running under init(1M)
Running vipca(silent) for configuring nodeapps
The given interface(s), "eth0" is not public. Public interfaces should be used
to configure virtual IPs.


# you MUST run vipca in the LAST NODE (NODE2) before clicking OK in OUI
# from $ORA_CRS_HOME/bin
cd /u01/crs/bin
./vipca
# in vipca, click on vip address to auto fill
rac1    rac1-vip.mydomain.com   192.168.4.13
rac2    rac2-vip.mydomain.com   192.168.4.14

When the configuration is complete, the final result report will be:
Configuration ResultsThe VIP Configuration Assistant has successfully created
resource aplications for each cluster node
Click 'Exit'

Click 'OK' in the OUI.
All the checks should succeed.
Click 'Exit'

# To verify the installation:
-- 1
cd /u01/stage10g/clusterware/cluvfy
./runcluvfy.sh stage -post crsinst -n rac1,rac2 -verbose
```

```
-- 2
ping rac1-vip
ping rac2-vip
-- 3
/u01/crs/bin/crsctl check crs

# To Avoid Node Eviction
# In ALL NODES as root: (IMPORTANT)
vi /u01/crs/install/rootconfig
At line 356, change
CLSCFG_MISCNT="-misscount 60"
to
CLSCFG_MISCNT="-misscount 300"
# and then by the command
crsctl set css misscount 300

# check status of crs daemon processes (ALL NODES)
./crs_stat -t
Name           Type           Target    State     Host
------------------------------------------------------------
ora.rac1.gsd   application    ONLINE    ONLINE    rac1
ora.rac1.ons   application    ONLINE    ONLINE    rac1
ora.rac1.vip   application    ONLINE    ONLINE    rac1
ora.rac2.gsd   application    ONLINE    ONLINE    rac2
ora.rac2.ons   application    ONLINE    ONLINE    rac2
ora.rac2.vip   application    ONLINE    ONLINE    rac2
# if State of any of them is UNKNOWN, try restarting the deamons
./crsctl stop crs
# then start again
./crsctl start crs
# to check their status (keep checking till all damons are up)
./crsctl check crs
# when they are all up, the output should be:
CSS appears healthy
CRS appears healthy
EVM appears healthy
```

3. Oracle Database 10g Software Installation

```
# make sure all clusterware processes are running
/u01/crs/bin/crs_stat -t
# you can restart them
/u01/crs/bin/crsctl stop crs
/u01/crs/bin/crsctl start crs

# If not loaded, to load the keys in the memory:
exec /usr/bin/ssh-agent $SHELL
/usr/bin/ssh-add

-- extract DB software in /u01/stage10g/database in node1
-- as oracle
mkdir /u01/stage10g/database

-- start OUI
cd /u01/stage10g/database
./runInstaller

Welcome 'Next' to skip the Welcome screen
```

```
Select Installation Type
'Next' to continue.

Install Location
Keep the default Oracle Base location /u01/app/oracle and
default Oracle Home location /u01/app/oracle/product/10.2.0/db_1
'Next' to continue.

Specify Hardware Cluster Installation Mode
Keep the default 'Cluster Installation' selection
and select both nodes.
'Next' to continue.

Product-Specific Prerequisite Checks
The OUI will now verify that the environment meets all the requirements. All
pre-requisite steps should complete successfully.
Select 'Next' to continue.

Select Configuration Option
Select 'Install Software Only'.
'Next' to continue.

Privileged Operating System Groups
Keep the default options dba, oinstall.
'Next' to continue

Summary
Review the summary and select 'Install' to start the installation.

Configuration Script
Once the installation is complete you will be prompted to run a script as the
root user. Open a
terminal window and execute the script as the root user on each node.
Select 'Ok' to continue after the script has been run successfully on both
nodes.

End of Installation
Once the installation is complete select 'Exit' to complete the installation
and exit the OUI.
```

4. Apply Patchset 3 (10.2.0.4) for Clusterware and Database Software

Note: This is the patchset applied for this environment. Generally speaking, if there is a newer version, use it instead of this version.

Note: This patch includes the Oracle Clusterware Process Monitor Daemon (oprocd) which when it detects a system hang, it restarts the hung node. This may restart a node under heavy workload. Check Oracle Clusteware Installation documenation for more information.

```
# extract 10g Release 2 (10.2.0.4) Patch Set 3 for Linux x86 to
/u01/stage10g/patch10.2.0.4/Disk1
mkdir /u01/stage10g/patch10.2.0.4

# If not loaded, to load the keys in the memory:
exec /usr/bin/ssh-agent $SHELL
/usr/bin/ssh-add

# you apply the patch on (1) clusterware then on (2) database software
```

```
# (1) Apply the patch on clusterware
# as oracle in rac1
cd /u01/stage10g/Patch10.2.0.4
./runInstaller
Select CRS Home then ->Next
Cluster Info displayed -> Next
Checks -> Next
->Install
After Installation is complete it will ask to apply a script on the nodes one-
by-one.
Following is a sample of the message that appears:
To complete the installation of this patchset, you must perform the following
tasks on each node:
 1. Log in as the root user.
 2. As the root user, perform the following tasks:
    a. Shutdown the CRS daemons by issuing the following command:
        /u01/crs/bin/crsctl stop crs
    b. Run the shell script located at:
        /u01/crs/install/root102.sh
     This script will automatically start the CRS daemons on the patched node
upon completion.
 3. After completing this procedure, proceed to the next node and repeat.

# following is the implementation of the above
su -
cd /u01/crs/bin
./crsctl stop crs
cd /u01/crs/install
./root102.sh
cd /u01/crs/bin
./crsctl query crs softwareversion
# REPEAT IN NODE2
# then check status of deamons
./crs_stat -t

# (2) Apply the patch on DB software
# as oracle in node1
cd /u01/stage10g/patch10.2.0.4/Disk1
./runInstaller
Select path of  ORACLE_HOME ->Next
After Installation is complete it will ask to apply a script on both the nodes
one-by-one.
```

5. Install EM Agent in cluster nodes (if required)

```

```

6. Configure Listeners

```
# Run netca (not netmanager which is not clusteraware) from any node.
# as oracle
cd $ORACLE_HOME/bin
./netca &
Select Cluster configuration and select both nodes.
Configure Listener
Then Configure Naming Methods: Local and Easy Connect

```

```
After installing listener check with below commands whether listener working
properly.
# as root
/u01/crs/bin/crs_stat -t
```

7. Perform ASM installation

```
# as oracle (in node1)
cd /u01/app/oracle/product/10.2.0/db_1/bin
dbca &

Welcome
Keep the default selection Oracle RAC database. 'Next' to continue.

Operations Select
Configure ASM. 'Next' to continue.

Node Selection
Select ALL the nodes and 'Next' to continue.

Create ASM Instance
Select a SYS password for the ASM instance.
Select IFILE parameter file to create. 'Next' to continue.

Select OK to confirm creation of the ASM instances.

ASM Disk Groups
Select Create New to create new ASM disk groups.

Enter dg1 as the first disk group name.
Keep the default redundancy settings (Normal) and select the 2 data disks
DISK1 and DISK2.
'Ok' to continue
Select Create New again to add another disk group.
Enter dg2 as the disk group name.
This time select External for redundancy and select the remaining disk DISK3.
This group will be used as the recovery arae
'OK' to continue.
All the disk groups are now created. Finish to complete.

# An ASM instance will be created in every node named as ASMn
export ORACLE_SID=ASM1
sqlplus /nolog
SQL>conn / as sysdba
select name from v$asm_diskgroup ;
...
# after checking, return ORACLE_SID to its original value
export ORACLE_SID=rac1
```

8. Perform cluster database creation

```
# If not loaded, to load the keys in the memory:
exec /usr/bin/ssh-agent $SHELL
/usr/bin/ssh-add

# as oracle (in node1)
cd /u01/app/oracle/product/10.2.0/db_1/bin
dbca &
```

```
Create Oracle RAC Database
'Next' to continue

Operations
Select Create a Database.
'Next' to continue.

Node
Select all the nodes
'Next' to continue.

Database Templates
Select the required template
'Next' to continue.

Database Identification
Enter rac as the global database name.
'Next' to continue.

Management Options
Keep the default settings Configure Enterprise Manager and Configure Database
Control for local management selected,
but Enable Alert Notifications and Enable Daily Disk Backup to Recovery Area
deselected.
'Next' to continue.

Database Credentials
Select Use the same Administrative Password for All Accounts and enter a
password.
'Next' to continue.

Storage Options
Select ASM for storage.
'Next' to continue.


ASM Disk Groups
Select both disk groups dg1 and dg2.
'Next' to continue.

Database File Locations
Keep the default Use Oracle-Managed Files.
Make sure +DG1 is entered as the Database Area.
'Next' to continue

Recovery Configuration:
Select Specify Flash Recovery Area and enter +DG2 as the Flash Recovery Area.
Set its size.
Select Enable archiving and click on the Edit Archive Mode Parameters button
and make sure the Flash Recovery area is pointing to +DG2.
'OK' and then
'Next' to continue.

Database Content
Select Sample Schemas if you want to install them.
Click 'Next'
```

```
Database Services:
Click Add button and enter Service Name such as: hrserv then click 'OK'
Make sure it's set to 'Preferred' in both nodes and select Basic for TAF.
Click 'Next'

Initialization Parameters:
Memory Size to 70%.  You can leave all other settings as is.
'Next' to continue

Creation Options
Select Generate database creation scripts if you want to review these at a
later stage.
'Finish' to review the installation.

Summary
Select Ok to close the review page and
'Finish' to start the installation.

Once database creation is done a summary screen will be displayed.
'Exit' to exit the OUI.

# check Oracle processes:
ps -eo pid -o command | grep ora_ | grep -v grep
```

9. Postinstallation tasks

```
As the oracle user edit the /etc/oratab file on both nodes.
Replace the database name with the instance name for the rac database
i.e. replace the rac keyword with rac1 or rac2 depending on the node.
Furthermore add details for your clusterware home to this file.
This will enable you to set the Clusterware home using the oraenv script.
Once edited the /etc/oratab file should contain the following:
vi /etc/oratab
On node RAC1:
+ASM1:/u01/app/oracle/products/10.2.0/db_1:N
rac1:/u01/app/oracle/products/10.2.0/db_1:N
crs:/u01/crs:N

On node RAC2
+ASM2:/u01/app/oracle/products/10.2.0/db_1:N
rac2:/u01/app/oracle/products/10.2.0/db_1:N
crs:/u01/crs:N


# verify the Cluster Registry configuration
srvctl config database -d rac

# backup the root.sh script
cp /u01/app/oracle/product/10.2.0/db_1/root.sh ~/root.sh.bak

# Back up the voting disk
dd if=/dev/raw/raw2 of=~/vdisk.bak

# verify that OEM is working
https://rac1.mydomain.com:1158/em
# restart the dbconsole if required
emctl status dbconsole
emctl stop dbconsole
```

```
emctl start dbconsole
```

10. Useful Postinstallation Tasks

- Following are tips to consider after the successful installation to make managing RAC easier.
- Consider using rlwrap utility with SQL*Plus and RMAN:
    - Using rlwrap Utility with RMAN in Unix-Based Systems
    - Using rlwrap Utility with SQL*Plus in Unix-Based Systems

```
/* Make crs_stat -t more readable */
/*  copy the following script into ~/scripts/crstat.sh */
#!/usr/bin/ksh
#
# Sample 10g CRS resource status query script
#
# Description:
#    - Returns formatted version of crs_stat -t, in tabular
#      format, with the complete rsc names and filtering keywords
#   - The argument, $RSC_KEY, is optional and if passed to the script, will
#     limit the output to HA resources whose names match $RSC_KEY.
# Requirements:
#   - $ORA_CRS_HOME should be set in your environment
# suggested scrip name: crstat.sh

RSC_KEY=$1
QSTAT=-u
AWK=/usr/bin/awk    # if not available use /usr/bin/awk

# Table header:echo ""
$AWK \
  'BEGIN {printf "%-45s %-10s %-18s\n", "HA Resource", "Target", "State";
          printf "%-45s %-10s %-18s\n", "-----------", "------", "-----";}'

# Table body:
$ORA_CRS_HOME/bin/crs_stat $QSTAT | $AWK \
 'BEGIN { FS="="; state = 0; }
  $1~/NAME/ && $2~/'$RSC_KEY'/ {appname = $2; state=1};
  state == 0 {next;}
  $1~/TARGET/ && state == 1 {apptarget = $2; state=2;}
  $1~/STATE/ && state == 2 {appstate = $2; state=3;}
  state == 3 {printf "%-45s %-10s %-18s\n", appname, apptarget, appstate;
state=0;}'


# then add the following in the .bashrc of oracle user
# if the file was saved in ~/scripts/crstat.sh
alias crstat='~/scripts/crstat.sh'
```

```
/* Easy Acces to crs and db homes */
# it is common to access bin directories in clusterware and db homes

# add the following to .bashrc of oracle user
alias db='cd $ORACLE_HOME/bin'
alias crs='cd $ORA_CRS_HOME/bin'
```

Installing Oracle 11g R2 RAC on Enterprise Linux 5

Note: The metalink document RAC Starter Kit and Best Practices (Linux) [ID 811306.1] is a good source reference for this task.

Main Changes in Oracle 11g Release 2 RAC

- SCAN: single client access name (SCAN) is a domain name used by all clients connecting to the cluster. It is registered to three IP addresses, either in the domain name service (DNS) or the Grid Naming Service (GNS).
- GNS: enables using DHCP, which must be configured in the subdomain in which the cluster resides.
- OCR and Voting can now be configured on an ASM diskgroup.
- Passwordless automatic SSH connectivity: the installer can configure SSH for you.
- Intelligent Platform Management interface (IPMI): provides a set of common interfaces to computer hardware and firmware that administrators can use to monitor system health and manage the system.
- Time sync: The new Oracle Cluster Time Synchronization Service is designed for organizations whose Oracle RAC databases are unable to access NTP services.
- Grid Infrastructure home: Clusterware and ASM share the same Oracle Home.
- Hangchecktimer replaced by the cluster synchronization service daemon Agent and Monitor to provide more accurate recognition of hangs and to avoid false termination.

Installation Environment

- Emulation software: VMWare Workstation 7
- RAC Nodes: 2 nodes with 2.5 GB RAM each , 2 ethernet cards.
- OS: Oracle Linux Enterprise 5 for x86 32-bit

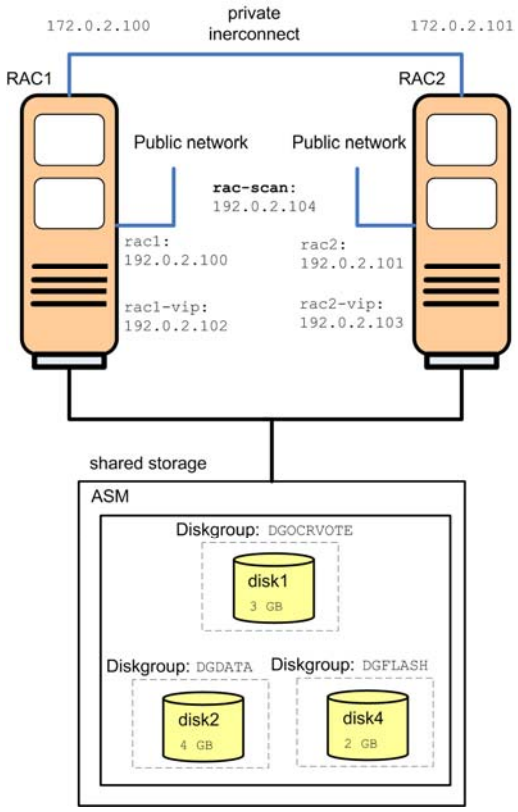
Required Software

- Oracle Database 11g Release 2 for Linux x86 32-bit
- Oracle Database 11g Release 2 Grid Infrastructure (11.2.0.1.0) for Linux x86 32-bit

Used Hardware

- In the VMWare: create one virtual machine (rac1) with the following specs:
  - 2.5 GB RAM
  - Two ethernet cards: both can be configured as bridged or host-only in VMware.
  - One local hardisk with 24 GB on SCSI 0:0.
  - CPU Count: 2

- Create a folder in the same directory structure level as the parent folder containing the created virtual machine. Give it a meaningful name like 'shared_disks'. Create in that folder the following disks:
    - Disk1: of 3 GB. Allocate its disk space. It will be used for OCR and Voting disk. Set it on controller SCSI 1:1.
    - Disk2: of 4 GB. Allocate its disk space. It will be used for +Data. Set it on controller SCSI 1:2.
    - Disk3: of 2 GB. Allocate its disk space. It will be used for +Flash. Set it on controller SCSI 1:3.



Installation Plan

11. Preinstallation tasks
    - Hardware requirements
    - Software requirements
    - Environment configuration
12. Oracle Grid Infrastructure installation
13. Oracle Grid Infrastructure Patching
14. Oracle Database 11g R2 Software Installation
15. Oracle Database 11g R2 Software Patching
16. Install EM Agent in cluster nodes (if required)

17. ASM Diskgroups Creation
18. RAC Database Creation
19. Complete postinstallation tasks
20. Useful postinstallation tasks

Note: The installation is explained without GNS and IPMI

Note: For this installation we will be using ASM for Clusterware and Database storage

1. Preinstallation tasks

- Install Oracle Enterprise Linux in the first local hardisk. Install nothing in the remaining disks.

  Note: for a production system, consider becoming an Oracle Unbreakable Linux customer and register your server on the Unbreakable Linux Network.

  - Configure the swap area in the local hardisk to have 3 GB disk space.
  - Give the first ethernet card IP 192.0.2.100 and the second 172.0.2.100 and the hostname `rac1.mydomain.com`. Define a gateway. If it does not exist, make it same as the host IP address.
  - Insall the following packages:
    - Desktop Environments
      - GNOME Desktop Environment
    - Applications
      - Graphical Internet (optional)
      - Editors (optional)
    - Development
      - Development Libraries
      - Development Tools
    - Servers
      - Do not select anything in this group.
    - Base System
      - Administration Tools
      - System Tools
        - Add the package 'sysstat' by clicking on the Details link and selecting "sysstat - The sar an iostat system monitoring commands." from the Optional Packages list.
    - X Window System
- Complete the installation.
- After the Installation compelets, RHEL 5.2 and below will hang on booting when it reaches to "starting udev" line. To solve this problem, shutdown the Vmware machine and change the CPU count and Core Count to only one. Implement the changes below, then shutdown the machine, set CPU count back to 2 and startup the machine.

  put the kernel command line parameters at the end of the "kernel" line:

  ```
  vi /boot/grub/grub.conf
  ```

  add `divider=10 clocksource=acpi_pm`

For example: `kernel /vmlinuz-2.6.18 .. clock=acpi_pm divider=10`

- For Vmware machines, install VMWare tools and set it to synchronize its time with the guest: `vmwaretoolbox`. Alternatvily, you can use Oracle Cluster Time Synchronization Service (ctssd) (metalink document 551704.1)
- Install further packages:

```
# to know distribution and version of Linux (Red Hat Ent. 5.2 used)
cat /etc/issue
# to know kernel version (and its errata level) (2.6.18-92 or newer)
uname -r

# to list missed packages:
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n' binutils \
compat-libstdc++-33 \
elfutils-libelf \
elfutils-libelf-devel \
gcc \
gcc-c++ \
glibc \
glibc-common \
glibc-devel \
glibc-headers \
ksh \
libaio \
libaio-devel \
libgcc \
libstdc++ \
libstdc++-devel \
make \
sysstat \
unixODBC \
unixODBC-devel

# for missed packages, install them:
rpm -Uvh libaio-devel-0.3.106-3.2.i386.rpm
rpm -Uvh unixODBC*

# Download the appropriate ASMLib RPMs from OTN.
# to know the kernel verion: uname -rm
# In this case we need:
oracleasm-2.6.18-92.el5-2.0.5-1.el5.i686.rpm
oracleasmlib-2.0.4-1.el5.i386.rpm
oracleasm-support-2.1.3-1.el5.i386.rpm

rpm -Uvh oracleasm*.rpm

# SELINUX must be disabled
cat /etc/selinux/config | grep SELINUX=
vi /etc/selinux/config
SELINUX=disabled
shutdown -h now -r
```

- Check the hardware requirements

```
# Hardware Requirements (in cluster nodes)
# At least 1.5 GB of physical memory but practically 1.5 is not fine
grep MemTotal /proc/meminfo
```

```
# swap space: same as the amount of physical memory
grep SwapTotal /proc/meminfo

# to display swap and memory in one command:
free

# if you don't have enought swap,
#  you can add swap space by creating a temporary swap file.
#  let's say about 500MB:
dd if=/dev/zero of=tempswap bs=1k count=500000
chmod 600 tempswap
mke2fs tempswap
mkswap tempswap
swapon tempswap

# 1 GB disk space in /tmp
df -h /tmp
# 8 GB of disk space for Oracle software
df

The size of the shared memory should be at least the greater of
MEMORY_MAX_TARGET and MEMORY_TARGET for each Oracle instance on the computer.
To determine the amount of shared memory available, enter the following
command:
# df -h /dev/shm/
```

- Create the required network configuration (rac2 will be created later):
  - Public and Private interface names must be the same for all nodes.
  - This private hostname does not need to be resolvable through DNS and should be entered in the /etc/hosts file.
  - SCAN VIPs must NOT be in the /etc/hosts file, it must be resolved by DNS. But here I've defined it as a single IP address in the "/etc/hosts" file, which is wrong and will cause the cluster verification to fail, but it allows me to complete the install without the presence of a DNS.
  - If you are using a DNS, Oracle recommends that you add lines to the /etc/hosts file on each node, specifying the public IP, VIP and private addresses.
  - If you configured the IP addresses in a DNS server, then, as the root user, change the hosts search order in /etc/nsswitch.conf on all nodes as shown:

    Old: `hosts: files nis dns`

    New: `hosts: dns files nis`

  - Then restart nscd daemon on each node: `/sbin/service nscd restart`

```
# Network names Resolution
# configure /etc/hosts if no domain server is used (both nodes)
vi /etc/hosts
127.0.0.1       localhost.localdomain   localhost
#eth0 - PUBLIC
192.0.2.100 rac1.mydomain.com rac1
192.0.2.101 rac2.mydomain.com rac2
#VIP
192.0.2.102 rac1-vip.mydomain.com rac1-vip
192.0.2.103 rac2-vip.mydomain.com rac2-vip
#eth1 - PRIVATE
```

```
172.0.2.100 rac1-priv
172.0.2.101 rac2-priv

# in real production: the follwing should not be there at all
# SCAN: cluster_name-scan.GNS_subdomain_name
192.0.2.104   rac-scan.mydomain.com rac-scan
```

- Create and configure the required OS users and groups

  Note: userid and groupid must be the same in all nodes. You can check them by `id oracle` command.

```
# all group and user ids on all the nodes must have identical id
# Grid Infrastructure (GI) and the Oracle RDBMS home will
# be installed using different users:
/usr/sbin/groupadd -g 501 oinstall
/usr/sbin/groupadd -g 502 dba
/usr/sbin/groupadd -g 504 asmadmin
/usr/sbin/groupadd -g 506 asmdba
/usr/sbin/groupadd -g 507 asmoper
/usr/sbin/useradd -u 501 -g oinstall -G asmadmin,asmdba,asmoper grid
/usr/sbin/useradd -u 502 -g oinstall -G dba,asmdba oracle

# set passwords
passwd oracle
passwd grid

# make sure nobody user exists (if not there, create it useradd nobody)
id nobody

# define the env variables for oracle user
vi /home/oracle/.bash_profile
# Oracle Settings
export EDITOR=vi
TMP=/tmp; export TMP
TMPDIR=$TMP; export TMPDIR
ORACLE_HOSTNAME=rac1.mydomain.com; export ORACLE_HOSTNAME
ORACLE_UNQNAME=RAC; export ORACLE_UNQNAME
ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
ORACLE_HOME=$ORACLE_BASE/product/11.2.0/db_1; export ORACLE_HOME
ORACLE_SID=RAC1; export ORACLE_SID
ORACLE_TERM=xterm; export ORACLE_TERM
PATH=/usr/sbin:$PATH; export PATH
PATH=$ORACLE_HOME/bin:$PATH; export PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export LD_LIBRARY_PATH
CLASSPATH=$ORACLE HOME/JRE:$ORACLE HOME/jlib:$ORACLE HOME/rdbms/jlib; export
CLASSPATH

# shell startup file
vi /etc/profile
if [ $USER = "oracle" ] || [ $USER = "grid" ]; then
 if [ $SHELL = "/bin/ksh" ]; then
  ulimit -p 16384
  ulimit -n 65536
 else
  ulimit -u 16384 -n 65536
 fi
 umask 022
fi
```

```
# for C shell
vi /etc/csh.login
if ( $USER = "oracle" || $USER = "grid" ) then
 limit maxproc 16384
 limit descriptors 65536
endif
```

- Configure kernel parameters and shell limits

  Note: If you make a mistake with a parameter setting and your system does not start, then you must start Linux in the single-user runlevel (runlevel 1). At this runlevel, the /etc/sysctl.conf file is not run.

```
# Kernel Parameters
# to tune thme, refer to metalink document 169706.1
# Append the following to the /etc/sysctl.conf file as the root user:
vi /etc/sysctl.conf

# kernel.shmmax not stated in 11g R2 (max: 4G) (169706.1)
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.aio-max-nr = 1048576
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576

# to take immediate effect
/sbin/sysctl -p

# User Shell Limits
# memlock is used to increase the per-process max locked memory
vi /etc/security/limits.conf
grid soft nproc 2047
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536

vi /etc/pam.d/login
session required pam_limits.so
```

- Create the required directories for the Oracle software:

```
# to know if there is an existing oracle inventory
# from its output, ORACLE_BASE will be parent of oraInventory
more /etc/oraInst.loc
# to identify existing Oracle home directories
more /etc/oratab

# Oracle Inventory Directory
# as a root
mkdir -p /u01/app/oraInventory
chown -R grid:oinstall /u01/app/oraInventory
chmod -R 775 /u01/app/oraInventory
```

```
# Grid Infrastructure Home Directory
mkdir -p /u01/11.2.0/grid
chown -R grid:oinstall /u01/11.2.0/grid
chmod -R 775 /u01/11.2.0/grid

# Oracle Base Directory
mkdir -p /u01/app/oracle
#needed to ensure that dbca is able to run after the rdbms installation
mkdir /u01/app/oracle/cfgtoollogs
chown -R oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/app/oracle

# Oracle RDBMS Home Directory
mkdir -p /u01/app/oracle/product/11.2.0/db_1
chown -R oracle:oinstall /u01/app/oracle/product/11.2.0/db_1
chmod -R 775 /u01/app/oracle/product/11.2.0/db_1
```

- Shutdown the Vmware machine then edit the VMware file (with vmx extensions) and add the following entry to allow sharing the disks (make sure the scsi controller number is the one you used):

```
disk.locking = "FALSE"
diskLib.dataCacheMaxSize = "0"
diskLib.dataCacheMaxReadAheadSize = "0"
diskLib.dataCacheMinReadAheadSize = "0"
diskLib.dataCachePageSize = "4096"

scsi1.sharedBus = "virtual"

scsi1:1.deviceType = "disk"
scsi1:2.deviceType = "disk"
scsi1:3.deviceType = "disk"
```

- Startup the machine then partition the disks:

Note: On a real life storage, you would create a single whole-disk partition with exactly 1 MB offset on each LUN to be used as ASM Disk. In fdisk: u (to change units from cylinder to sectors), n, p, 1, 2048, w.

```
# as a root, for the disks /dev/sdb .. /dev/sdd
# confirm they are seen:
ls /dev/sd*
#partition the disks:
fdisk /dev/sdb
# answers: "n", "p", "1", "Return", "Return", "p" and "w"
Note: if the following message appears after the "w" command:
 WARNING: Re-reading the partition table failed with error 16: Device or
 resource  busy, then you can avoid restarting the machine by the following
command: partprobe

# to make sure partions are created
ls -lX /dev/sd*
```

- Configure ASM drivers:

  Note: If you see that the shared disks are not synced between rac1 and rac2, one of the things you can examine is to see if there is any "debug" command in any of the nodes' vmx files. If you find one, shutdown the node, remove the command from the vmx file and restart.

```
# as root  (to be done in all nodes)
```

```
oracleasm configure -i

Default user to own the driver interface []: grid
Default group to own the driver interface []: asmdba
Start Oracle ASM library driver on boot (y/n) [n]: y
Fix permissions of Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration:           [  OK  ]
Creating /dev/oracleasm mount point:                       [  OK  ]
Loading module "oracleasm":                                [  OK  ]
Mounting ASMlib driver filesystem:                         [  OK  ]
Scanning system for ASM disks:                             [  OK  ]

# Load the kernel module using the following command:
/usr/sbin/oracleasm init
# If you have any problems, make sure you have the correct
#  version of the driver:
/usr/sbin/oracleasm update-driver

# mark the shared disks: (one node)
/usr/sbin/oracleasm createdisk DISK1 /dev/sdb1
/usr/sbin/oracleasm createdisk DISK2 /dev/sdc1
/usr/sbin/oracleasm createdisk DISK3 /dev/sdd1

# check the disks are marked and seen:
/usr/sbin/oracleasm listdisks

# in other nodes:
/usr/sbin/oracleasm scandisks
/usr/sbin/oracleasm listdisks

#If you need to unmark a disk that was used in a createdisk command:
/usr/sbin/oracleasm deletedisk DISK1
/usr/sbin/oracleasm deletedisk DISK2
/usr/sbin/oracleasm deletedisk DISK3
```

- Disable screensavers on host & guest machines.
  - In Oracle Linux: Applications-> Preferences-> Screen Saver
  - Do the same after logging off and logging on again as `oracle` and `grid` user.
- Shutdown rac1
- Copy the folder containing rac1 into a new folder in the same directory structure level. Let's name it "rac2". This will be the second node in the cluster.
- Edit the VMware file of rac1 and edit the following: displayName = "rac2"
- Open rac2, then perform:
  - in a terminal issue: system-config-network-gui.
  - Remove the devices with the "%.bak" nicknames. To do this, highlight a device, deactivate, then delete it.
  - Highlight the "eth0" interface and click the "Edit" button. Change its IP addresses and gate way: IP 192.0.2.101. Click on the "Hardware Device" tab and click the "Probe" button.
  - For eth1 set its ip address to 172.0.2.101. Do not define a gateway.
  - In DNS tab, change hostname to rac2.mydomain.com.
  - Activate the network cards.

- In rac2, perform:

```
# change the variable in the file
vi /home/oracle/.bash_profile
ORACLE_SID=RAC2; export ORACLE_SID
ORACLE_HOSTNAME=rac2.localdomain; export ORACLE_HOSTNAME
```

- Start rac1. Make sure the machines can see each other:

```
ping -c 3 rac1
ping -c 3 rac1-priv
ping -c 3 rac2
ping -c 3 rac2-priv
```

2. Oracle Grid Infrastructure installation

```
# in rac1: copy the software in a staging folder
mkdir -p /u01/app/stage/ora11gr2gridinfra
chown -R grid:oinstall /u01/app/stage/ora11gr2gridinfra
chmod -R 775 /u01/app/stage/ora11gr2gridinfra

mkdir -p /u01/app/stage/ora11gr2db
chown -R oracle:oinstall /u01/app/stage/ora11gr2db
chmod -R 775 /u01/app/stage/ora11gr2db

# in all nodes, make sure the asm disks are accessible:
/usr/sbin/oracleasm scandisks
/usr/sbin/oracleasm listdisks

# do not use cluvfy because SSH was not configured.

# lunch OUI from the clusterware ( as grid from rac1)
# if logged in in Genome using another user, log out and log in as grid
cd /u01/app/stage/ora11gr2gridinfra
./runInstaller

Installation Option
>Select radio button 'Install and Configure Grid Infrastructure for a Cluster'
>Next

Installation Type
>Select 'Advanced Installation'
>Next

Product Language
>Accept 'English' as language'
>Next

Grid Plug and Play
>cluster name: rac
>SCAN name:rac-scan.mydomain.com
>Make sure 'Configure GNS' is NOT selected
>Next

Cluster Node Information
>Add button
>Hostname:rac2.mydomain.com
>Virtual IP Name: rac2-vip.mydomain.com
>OK
```

```
>"SSH Connectivity" button
>Enter the password
>Setup button
>Test button

Network Interface Usage
>check the public and private networks are specified correctly
>Next

Storage Option
>Select 'Automatic Storage Management (ASM)'
>Next

Creat ASM Disk Group
>Disk Group Name: DGOCRVOTE (3GB disk: Disk1)
>Redundancy: external
>Next
NOTE: If you see an empty screen for you candidate disks it is likely that
ASMLib has not been properly configured. Try reconfigure them.
If you are sure that ASMLib has been properly configured click on 'Change
Discovery Path' and provide the correct destination.

ASM Password
>Specify and conform the password you want to use
>Next

Failure Isolation Support
>Select NOT to use IPMI
>Next

Privileged OS Groups
>Assign the correct OS groups for OS authentication (mostly default is OK)
>Next

Installation Location
>ORACLE_BASE: /u01/app/oracle
 Software location: /u01/11.2.0/grid
>Next

Create Inventory
>Specify the locations: /u01/app/oraInventory
>Next

Perform Prerequisite Checks
>OUI performs certain checks
>Check that status of all checks is Succeeded
Note: in this example, NPS error can be ignored
>Next

Summary
>Finish

Execute Configuration Scripts
>Run the scripts as instructed in the screen
Note: The scripts must be run on one node at a time.
>OK
```

```
We expect the verification phase to fail with an error relating to the SCAN,
assuming you are not using DNS.
    INFO: Checking Single Client Access Name (SCAN)...
    INFO: Checking name resolution setup for "rac-scan.localdomain"...
    INFO: ERROR:
    INFO: PRVF-4664 : Found inconsistent name resolution entries for SCAN name
"rac-scan.localdomain"
    INFO: ERROR:
    INFO: PRVF-4657 : Name resolution setup check for "rac-scan.localdomain"
(IP address: 192.168.2.201) failed
    INFO: ERROR:
    INFO: PRVF-4664 : Found inconsistent name resolution entries for SCAN name
"rac-scan.localdomain"
    INFO: Verification of SCAN VIP and Listener setup failed
Provided this is the only error, it is safe to ignore this
>Next

Message: The installation of the Grid Infrastructure was successfull.
>Close

Note: If your OS is SUSE Linux, shutting down on node will result in shutting
the other nodes. To workaround:
#cd /etc/rc3.d
#ln -s /etc/init.d/ohasd K07ohasd
```

3. Oracle Grid Infrastructure Patching

```
Apply patch set, if there is any.
```

4. Oracle Database 11g R2 Software Installation

```
# make sure all clusterware processes are running
/u01/crs/bin/crs_stat -t

# as oracle
./runInstaller

Configure Security Updates
>Provide your e-mail address, if you want
>Next

Installation Options
>Select 'Install Database software only'
>Next>

Install Type
>Select 'Real Application Clusters database installation', and select all
nodes.
>Use the 'SSH Connectivity' button to configure/test the passwordless SSH
connectivity.
>Next

Product Languages
>Confirm 'English'
>Next

Database Edition
```

```
>'Enterprise Edition' is ticked
>Next

Installation Location
>Oracle Base: /u01/app/oracle
Software Location: /u01/app/oracle/product/11.2.0/db_1
>Next

Privileged OS Groups
>OSDBA: dba
>OSOPER: oinstall
>Next

Prerequisite Checks
..OUI performs prerequisite checks
>Check that status of all checks is Succeeded
>If you are sure the unsuccessfull checks can be ignored tick the box 'Ignore
All'
>Next

Summary
>Check summary info
>Finish

Install Product
..OUI installs the db software
>as a root, run the root.sh script on the first node then the other nodes (One
at a time)
>OK

Finish
>Close
```

5. Oracle Database 11g R2 Software Patching

```
```

6. Install EM Agent in cluster nodes (if required)

```
```

7. ASM Diskgroups Creation

Note: It is Oracle's Best Practise to have an OCR mirror stored in a second disk group. To follow this recommendation add an OCR mirror. Mind that you can only have one OCR in a diskgroup. To add OCR mirror to an Oracle ASM disk group, ensure that the Oracle Clusterware stack is running and

```
ocrconfig -add +ORADATA
```

```
ocrcheck
```

```
# as grid user: start the ASM Configuration Assistant (ASMCA)
#su - grid
cd /u01/11.2.0/grid/bin
./asmca

>Disk Groups tab
>Create button
>Disk Group Name: DGDATA
```

```
>Redundancy: External
>Disk2
>OK

>Create button
>Disk Group Name: DGFLASH
>Redundancy: External
>Disk3
>OK

>Exit
>Yes
```

8. RAC Database Creation

```
# as oracle
cd /u01/app/oracle/product/11.2.0/db_1/bin
./dbca

Welcome
Select 'Oracle Real Application Clusters database'
>Next

Operations
> choose option 'Create a Database'
>Next

Database Template
>Select General Purpose or any template
>Next

Database Identification
>Configuration Type: Admin
>Globale Database Name: rac
>SID: rac
>Select All button
>Next

Management Options
>Select the option you want
>Next

Database Credentials
>Set the password(s)
>Next

Database File Locations
>Database Area: +DGDATA
>Practically (but not in this case), you should define 'Multiplex Redo Logs
and Control Files'.
>Next

/* Note: If you cannot see the diskgroups, perform the following (ID:
1177483.1):
   su -
   cd <Grid_Home>/bin
   chmod 6751 oracle
   ls -l oracle
```

```
   -rwsr-s--x 1 grid oinstall
*/

ASM Credentials
..If you chose to set up EM, you will be asked about ASMSNMP password
>Enter the password
>Ok button

Recovery Configuration
>Flash recovery area: +DGFLASH
>define the size: 2000 MB
 If the size is smaller than recommended a warning will popup.
>Enable Archiving
>Next

Database Content
>Select if you want to have sample schemas created in your database
>Next

Initialization Parameters
>Review and change the settings for memory allocation, characterset etc.
>Next

Database Storage
>Review the database storage settings and change as required
>Next

Creation Options
>Make sure the tickbox 'Create Database' is ticked
>Finish

Summary
>OK
.. Database creation proceeding
>after completion Exit

# Confirmation

# to show the current configuration and status of the RAC database
srvctl config database -d rac

# check OEM (if configured):
https://rac1.mydomain.com:1158/em/
# if not started, you can start it:
su - oracle
cd /u01/app/oracle/product/11.2.0/db_1/bin
export ORACLE_UNQNAME=rac
./emctl status dbconsole

# check Oracle processes:
ps -eo pid -o command | grep ora_ | grep -v grep
```

9. Postinstallation tasks

```
# backup the root.sh script (on all nodes)
cp /u01/app/oracle/product/11.2.0/db_1/root.sh ~/root.sh.bak
```

10. General Useful Postinstallation Tasks in Linux

- Following are tips to consider after the successful installation to make managing RAC easier.
- Consider using rlwrap utility with SQL*Plus and RMAN:
    - Using rlwrap Utility with RMAN in Unix-Based Systems
    - Using rlwrap Utility with SQL*Plus in Unix-Based Systems

```
/* Make crs_stat -t more readable */
/*  copy the following script into ~/scripts/crstat.sh */
#!/usr/bin/ksh
#
# Sample 10g CRS resource status query script
#
# Description:
#    - Returns formatted version of crs_stat -t, in tabular
#      format, with the complete rsc names and filtering keywords
#   - The argument, $RSC_KEY, is optional and if passed to the script, will
#     limit the output to HA resources whose names match $RSC_KEY.
# Requirements:
#   - $ORA_CRS_HOME should be set in your environment
# suggested scrip name: crstat.sh

RSC_KEY=$1
QSTAT=-u
AWK=/usr/bin/awk    # if not available use /usr/bin/awk

# Table header:echo ""
$AWK \
  'BEGIN {printf "%-45s %-10s %-18s\n", "HA Resource", "Target", "State";
          printf "%-45s %-10s %-18s\n", "-----------", "------", "-----";}'

# Table body:
$ORA_CRS_HOME/bin/crs_stat $QSTAT | $AWK \
 'BEGIN { FS="="; state = 0; }
  $1~/NAME/ && $2~/'$RSC_KEY'/ {appname = $2; state=1};
  state == 0 {next;}
  $1~/TARGET/ && state == 1 {apptarget = $2; state=2;}
  $1~/STATE/ && state == 2 {appstate = $2; state=3;}
  state == 3 {printf "%-45s %-10s %-18s\n", appname, apptarget, appstate;
state=0;}'


# then add the following in the .bashrc of oracle user
# if the file was saved in ~/scripts/crstat.sh
alias crstat='~/scripts/crstat.sh'
```

```
/* Easy Acces to crs and db homes */
# it is common to access bin directories in clusterware and db homes

# add the following to .bashrc of oracle user
alias db='cd /u01/app/oracle/product/11.2.0/db_1/bin'

# add the following to .bashrc of grid user
alias crs='cd /u01/app/oracle/crs/bin'
```

Installing Oracle 10g R2 RAC on Windows

Note: The metalink document RAC Starter Kit and Best Practices (Windows) [ID 811271.1] is a good source reference for this task.

Installation Methods

- NTFS_RAW_ASM (shown in this document)
- NTFS_RAW_RAM
- NTFS_OCFS_OCFS

Installation Environment

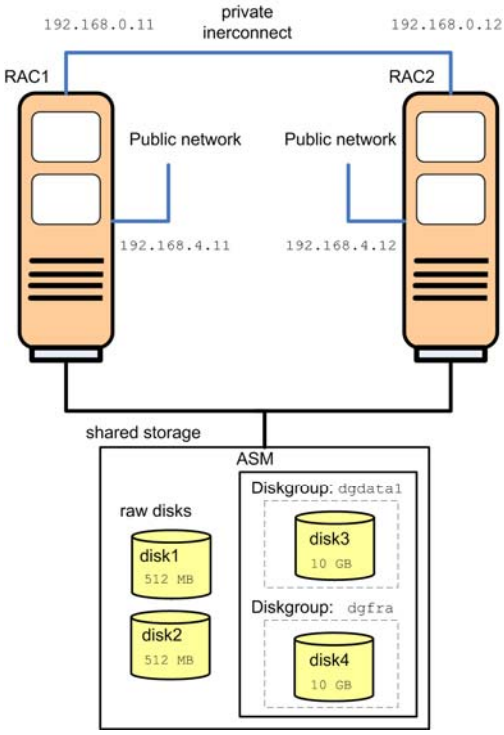
- Emulation software: VMWare Server 2 for Windows.
- RAC Nodes: 2 nodes
- OS: Windows 2003 Server Standard Edition 32-bit SP2

Required Software

- Oracle 10g R2 Clusterware for Windows 32-bit
- Oracle Database 10g Release 2 for Windows 32-bit

Used Virtual Hardware

- In the VMWare: create one virtual machine (rac1) with the following specs:
  - 2 GB RAM
  - two ethernet cards: both can be configured as bridged or host-only in VMware.
  - one local hardisk with 20 GB on controller SCISI 0:0.
  - CPU Count: 2
  - create a folder in the same directory structure level as the parent folder containing the created virtual machine. Give it a meaningful name like 'shared_disks'. Create in that folder two disks of 10 GB and two of 512 MB each. All of them are of LSI Logic type and SCISI Persistent. Make sure they are on SCISI controller different from the SCSI0. For example, set them on SCSI1.



Installation Plan

1. Preinstallation tasks
2. Oracle Clusterware installation
3. Apply Patch Set 3 (10.2.0.4) on Clusterware software
4. Oracle ASM 10g Software Installation
5. Apply Patchset 3 (10.2.0.4) on ASM software
6. Install EM Agent in cluster nodes (if required)
7. Configure Listeners
8. Create ASM Instance
9. Install Oracle RAC Database Home Software
10. Apply Patchset 3 (10.2.0.4) on Oracle RAC Software Home
11. Perform cluster database creation
12. Useful postinstallation steps

1. Preinstallation tasks

- The local admin username and password must be the same on both nodes.
- Give the first ethernet card IP 192.168.4.11 and the second 192.168.0.11. Define a gateway. If it does not exist, make it same as the host IP address.

  Note: Gateway must be defined and its IP must be alive.
- Set the hostname to `rac1.mydomain.com` (Desktop-> right click My Computer-> Properties-> Computer Name-> Change-> type computer name: rac1 -> More-> type Primary DNS Suffix: mydomain.com

```
/* Prepare the raw disks */
# enable raw disk auto mounting
diskpart
AUTOMOUNT ENABLE
exit
# then reboot

# start disk management
start-> run-> type: diskmgmt.msc-> Disk Initialization Wizard pops up->
Next->
mark all disks (1-4) to initialize->
make sure all disks are unmarked Next-> Finish

# all disks must be in Basic mode (not dynamic)

right click Disk1-> New Partition-> Next->
select the Extended partition radio button-> Next->
select the partition size to fill the disk-> Next-> Finish

right click on the partition and select the "New Logical Drive" -> Next->
Accept the default partition size Next->
select the "Do not assign a drive letter or drive path" option->
Next->
select the "Do not format this partition" option-> Next-> Finish

Repeat the previous partitioning steps for the remaining disks.


/* Time Sync */
-- make sure the time is synched in Vmwaretools
-- in Windows, in Date and Time Settings, there is an option "Automatically
syn with Internet Server" which can be configured, if connected to Web
```

- Create the required network configuration (rac2 will be created later):

```
# Network names Resolution
# if no domain server is used (both nodes)
notepad C:\WINDOWS\system32\drivers\etc\hosts
127.0.0.1       localhost
#Public
192.168.4.11    rac1.mydomain.com    rac1
192.168.4.12    rac2.mydomain.com    rac2
#VIP
192.168.4.13    rac1-vip.mydomain.com   rac1-vip
192.168.4.14    rac2-vip.mydomain.com   rac2-vip
#Inter-connect
192.168.0.11    rac1-priv.mydomain.com       rac1-priv
192.168.0.12    rac2-priv.mydomain.com       rac2-priv
```

- Disable Windows Media Sensing, which allows Windows to uncouple an IP address from a card when the link to the local switch is lost:

```
in the registry:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters
Add the following registry entry to the Parameters subkey:
 Name: DisableDHCPMediaSense
 Data type: REG_DWORD (Boolean)
 Value: 1
```

- Open the "Network Connections" screen (Start > All Programs > Accessories > Communications > Network Connections). Rename the two connections to "public" and "private" respectively, making sure you apply the names to the appropriate connections.

- Ensure the public interface is first in the bind order:
  - Open the "Network Connections" dialog by right-clicking on the "My Network Places" icon and selecting the "Properties" menu option.
  - Select the "Advanced > Advanced Settings..." menu option.
  - On the "Adapters and Bindings" tab, make sure the public interface is the first interface listed. Otherwise, promote it.
- Disable screensavers
- Shutdown rac1
- Edit the VMware file (with vmx extensions) and add the following entry to allow sharing the disks:

```
disk.locking = "FALSE"
diskLib.dataCacheMaxSize = "0"
diskLib.dataCacheMaxReadAheadSize = "0"
diskLib.dataCacheMinReadAheadSize = "0"
diskLib.dataCachePageSize = "4096"
diskLib.maxUnsyncedWrites = "0"

scsi1.sharedBus = "virtual"
scsi1:1.deviceType = "disk"
scsi1:2.deviceType = "disk"
scsi1:3.deviceType = "disk"
scsi1:4.deviceType = "disk"
scsi1:5.deviceType = "disk"
```

- Copy the folder containing rac1 into a new folder in the same directory structure level. Let's name it "rac2". This will be the second node in the cluster.
- Edit the VMware file of rac1 and edit the following: displayName = "rac2"
- Start rac2 then change it's IP addresses: IP 192.168.4.12 and 192.168.0.12. It's hostname to rac2.mydomain.com. Modify their default gateway accordingly.
- Change hostname to rac2 and restart the virtual machine.
- Start rac1 and make sure it can ping rac2.
- Perform Net Use test, to make sure files can be transfered:

```
# on rac1
net use \\rac2\c$
# on rac2
net use \\rac1\c$
```

Note: The environment variables TMP & TEMP must point to the same directory in all RAC nodes, which is the case here.

- Stage Oracle software (Clusterware, DB and patch set) in rac1.
- Run the CVU to check the state of the cluster prior to the install of the Oracle Software. The messages 'The system cannot find the file specified' and 'Could not find a suitable set of interfaces for VIPs' can be ignored.

```
C:\temp\OracleClusterware\cluvfy>runcluvfy stage -post hwos -n rac1,rac2
C:\..\cluvfy>runcluvfy stage -pre crsinst -n rac1,rac2 -verbose
```

2. Oracle Clusterware installation

- if you are using terminal services to perform the installation, be sure to invoke the terminal services in 'console' mode:

```
mstsc -v:servername /F /console
```

OR, for some (newer) versions of Windows:

```
mstsc -v:servername /F /admin
```

```
/* Stop Interrupting Services on both nodes */
# the MSDTC service may interrupt installation process
# it can be started after installation
stop the service: Distributed Transaction Coordinator

# lunch OUI from the clusterware software (on rac1)
cd C:\temp\OracleClusterware\
./setup.exe


>Welcome messge
>Next

>enter Home Details
OraCr10g
C:\oracle\product\10.2.0\crs

>Product-Specific Prerequisite Checks
 They all should succeed

>Specify Cluster Configuration
Select 'Add' and add the details for node rac2. Enter all details
>OK
rac2.mydomain.com
rac2-priv.mydomain.com
rac2-vip.mydomain.com
>Next

>Specify Network Interface Usage
set the Public interface (subet net 192.168.4.*)
>Ok

>Specify Oracle Cluster Registry (OCR) Location
>Highlight disk 1 and click the "Edit"
>select the "Place OCR(Primary) on this Partition" option >OK
>Highlight disk 2 and click the "Edit"
>Select the "Place Voting Disk on this Partition" option >OK
>Next and ignore the redundancy warnings
>OK

>Summary
>Install

>Wait while the configuration assistants run

If the Configuration Assistant fails, there is a problem that must be fixed
before proceeding. Metalink documents might help: 356535.1, 310791.1

VIPCA should fails, just click the "OK" button on the resulting error screen
>Next button and accept the subsequent warning
>Exit

On the RAC1 virtual machine, run the VIPCA manually:
cd c:\oracle\product\10.2.0\crs\bin
vipca.bat
```

```
>Welcome
>Next

>Highlight the "public" interface
>Next

>Enter the virtual IP alias and address for each node: rac1-vip.mydomain.com
Once you enter the first alias, the remaining values should default
automatically
>Next

>Summary
>Finish


# To verify the installation:
cd C:\oracle\product\10.2.0\crs\BIN
-- 1
cluvfy stage -post crsinst -n rac1,rac2
-- 2
ping rac1-vip
ping rac2-vip
-- 3
crsctl check crs

-- 4
# check status of crs daemon processes (ALL NODES)
crs_stat -t
Name           Type           Target    State     Host
------------------------------------------------------
ora.rac1.gsd   application    ONLINE    ONLINE    rac1
ora.rac1.ons   application    ONLINE    ONLINE    rac1
ora.rac1.vip   application    ONLINE    ONLINE    rac1
ora.rac2.gsd   application    ONLINE    ONLINE    rac2
ora.rac2.ons   application    ONLINE    ONLINE    rac2
ora.rac2.vip   application    ONLINE    ONLINE    rac2

# if State of any of them is UNKNOWN, try restarting the deamons
crsctl stop crs
# then start again
crsctl start crs
```

3. Apply Patch Set 3 (10.2.0.4) on Clusterware software

- There is a bug in Oracle Clusterware 10.2.0.1 in which sometime CRSS service is unable to access OCR disk. This issue is addressed by Patch Set 3. Therefore, to avoid any possible problem by this issue, you should apply the Patch Set at this stage.

```
# (1) Apply the patch on clusterware
# in ALL NODES
# Stop the following services  and make their startup type Manual (if
running):
Oracle Object Service
OracleClusterVolumeService
OracleCRService
OracleCSService
OracleEVMService
```

```
Open task manager and kill ons processes, if there is any.

# if any of the services is hang, you can change the startup type from regedit
then reboot (don't forget to stop the Distributed Transaction service after
reboot) :
HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Services ->
OracleCSService -> Select (Start) -> Edit -> Specify 3 in Value


# in RAC1: start oui
cd C:\temp\Patchset_10204\Disk1>
setup.exe
Welcome
>Next

>select Oracle Clusterware home
>Next
>Confirm nodes >Next
>Checking should pass >Next
>Summary
>Install
>Exit

# update the nodes
# make sure all related Oracle services are stopped, otherwise stop them
# apply the batch in node1 then node2:
C:\oracle\product\10.2.0\crs\install\patch102.bat
# it should end with the following message:
Successful upgrade of this node to Oracle Cluster Ready Services

# Oracle Process Manager service will automatically be installed

# check the updated verion
crsctl query crs activeversion

# then check status of deamons
crs_stat -t

# change the startup mode of the serives to AUTOMATIC
# startup Distributed Service, if it was stopped.
```

4. Oracle ASM 10g Software Installation

```
# make sure all clusterware processes are up and running
cd C:\oracle\product\10.2.0\crs\BIN
crs_stat -t

-- in rac1: start OUI
cd C:\temp\OracleDB10gR2
setup.exe

Welcome
>Next

Select Installation Type
>Next
```

```
Install Location
>Oracle Home name: OraDb10g
>Oracle Home location: C:\oracle\product\10.2.0\db_1
>Next

Specify Hardware Cluster Installation Mode
>select both nodes
>Next

Product-Specific Prerequisite Checks
All pre-requisite steps should complete successfully.
>Next

Select Configuration Option
>Install Software Only
>Next

Summary
>Finish

End of Installation
>Exit

Optionally, check and modify, if you wish, NLS_LANG in regedit.
```

5. Apply Patchset 3 (10.2.0.4) on ASM Software

Note: This is the patchset applied for this environment. Generally speaking, if there is a newer version, use it instead of this version. In all cases, same patch set applied to Oracle software must also be applied to Clusterware first.

```
#Apply the patch on DB software
# in RAC1: start oui
cd C:\temp\Patchset_10204\Disk1>
setup.exe
Welcome
>Next

>select Oracle Database home
>Next
>Confirm nodes >Next
>Checking should pass >Next
>Summary
>Install

>Exit

# not required, but recommended:
restart the nodes
```

6. Install EM Agent in cluster nodes (if required)

```

```

7. Configure Listeners

```
# Run netca (not netmanager which is not clusteraware) from any node
cd C:\oracle\product\10.2.0\db_1\bin
netca.bat
```

```
>Select Cluster configuration and select both nodes.
>Configure Listener
>Configure Naming Methods: Local and Easy Connect

After installing listener check with below commands whether listener working
properly
C:\oracle\product\10.2.0\crs\bin\crs_stat -t
```

8. Create ASM Instance

```
# in rac1
cd C:\oracle\product\10.2.0\db_1\bin
dbca.bat

Welcome
Keep the default selection Oracle RAC database
>Next

Operations Select
>Configure ASM >Next

Node Selection
>Select ALL the nodes >Next

Create ASM Instance
Select a SYS password for the ASM instance.
Select IFILE parameter file to create
>Next
>OK (asm instances will be created)

# here's an issue faced in a case after creating the ASM instances:
# check that the Administrator (or the OS you're using) belong to ora_dba
# group in ALL the Nodes. If not there, add the user to the group.


# Stamp the disks for ASM
>Create New
>Stamp Disks
>Select the "Add or change label" option
>Next
>select the first Disk of size 10G
>type DATA in the prefix text field (don't type ASM)
>Next >Next >Finish

>stamp the second 10g disk: repeat above for the second disk

rac2 should see the changes:
asmtool -list

Note: if you want to clean the disk to redo the procedure above, you can:
(1) in rac1: DISKPART, select <DISK NAMe>, clean all, create part ext, create
part log
(2) in rac2: remove letter drive assigned to the partition

# create ASM Disk Groups
>Enter dgdata1 as the first disk group name.
>Set redundancy settings to External
```

```
>select the disk ORCLDISKDATA0
>Ok

Select Create New again to add another disk group named as dgfra. This group
will be used as the recovery arae
>OK
>Finish to complete.

Note: in an environment, I kept receiving ORA-15063 error. After making sure
that all the pre-requisits were applied, the issue was resolved by allocating
all the disk space at disk creation time.

# An ASM instance will be created in every node named as ASMn
set ORACLE_SID=ASM1
sqlplus /nolog
SQL>conn / as sysdba
select name from v$asm_diskgroup ;
...
```

9. Install Oracle RAC Database Home Software

```
# make sure all clusterware processes are up and running
cd C:\oracle\product\10.2.0\crs\BIN
crs_stat -t


-- in rac1: start OUI
cd C:\temp\OracleDB10gR2
setup.exe

Welcome
>Next

Select Installation Type:
Enterprise Edition
>Next

Install Location
>Oracle Home name: OraDb10g2
>Oracle Home location: C:\oracle\product\10.2.0\db_2
>Next

Specify Hardware Cluster Installation Mode
>select both nodes
>Next

Product-Specific Prerequisite Checks
All pre-requisite steps should complete successfully
>Next

Upgrade an Existing Database
>select No
>Next

Select Configuration Option
>Install Software Only
>Next
```

```
Summary
>Finish

End of Installation
>Exit

Optionally, check and modify, if you wish, NLS_LANG in regedit.
```

10. Apply Patchset 3 (10.2.0.4) on Oracle RAC Software Home

```
#Apply the patch on DB software
# in RAC1: start oui
cd C:\temp\Patchset_10204\Disk1>
setup.exe
Welcome
>Next

>select Oracle RAC Database home
>Next
>Confirm nodes >Next
Oracle Configuration Manager
you can setup the Oracle Configuration Manager
>Checking should pass >Next
>Summary
>Install

>Exit
```

11. Perform cluster database creation

```
# in rac1: from Oracle Database home (not ASM)
c:\oracle/product/10.2.0/db_2/bin\dbca

Create Oracle RAC Database
'Next' to continue

Operations
Select Create a Database.
'Next' to continue.

Node
Select all the nodes
'Next' to continue.

Database Templates
Select the required template: like "General Purpose"
'Next' to continue.

Database Identification
Enter rac as the global database name.
'Next' to continue.

Management Options
Keep the default settings "Configure Database Control for Configure Enterprise
Manager" selected,
Keep "Enable Alert Notifications" and "Enable Daily Disk Backup to Recovery
Area" deselected.
'Next' to continue
```

```
Database Credentials
Select Use the same Administrative Password for All Accounts and enter a
password.
'Next' to continue

Storage Options
Select ASM for storage
'Next' to continue

ASM Disk Groups
Select the disk groups dgdata1
'Next' to continue

Database File Locations
Keep the default Use Oracle-Managed Files.
Make sure +DGDATA1 is entered as the Database Area.
'Next' to continue

Recovery Configuration:
Select Specify Flash Recovery Area and enter +DGFRA as the Flash Recovery
Area. Set its size.
Select Enable archiving and click on the Edit Archive Mode Parameters button
and make sure the Flash Recovery area is pointing to +DGFRA
'OK' and then
'Next' to continue.

Database Content
Select Sample Schemas if you want to install them.
Click 'Next'

Database Services:
Click Add button and enter Service Name such as: hrserv then click 'OK'
Make sure it's set to 'Preferred' in both nodes and select Basic for TAF.
Click 'Next'

Initialization Parameters:
Memory Size to 70%.  You can leave all other settings as is.
'Next' to continue

Database Storage
Here you can review the placement of various database files
'Next' to continue

Creation Options
Select Generate database creation scripts if you want to review these at a
later stage.
'Finish' to review the installation.

Summary
Select Ok to close the review page and
'OK' to start the installation.

Once database creation is done a summary screen will be displayed.
Copy the OEM URL into clipboard. Save it in a file or the Internet Browser
'Exit' to exit the OUI.
```

12. Useful Postinstallation Steps

- Following are tips to consider after the successful installation to make managing RAC easier.

```
# Create links to Oracle crs and database homes.
notepad  C:\WINDOWS\system32\crs.bat
cd /d C:\oracle\product\10.2.0\crs\BIN

notepad  C:\WINDOWS\system32\db.bat
cd /d C:\oracle\product\10.2.0\db_2\BIN

notepad  C:\WINDOWS\system32\asm.bat
cd /d C:\oracle\product\10.2.0\db_1\BIN
```

Cleaning Up Clusterware Installation on Windows

- If clusterware installation fails, sometimes you need to clean up the installation and redo the installation again. Following procedure just does that. The procedure applies on Oracle 10g R1, R2 and Oracle 11g R1.

```
1. Stop the following services on each node and set them to Manual:
 OracleCSService
 OracleCRService
 OracleEVMService


2. To see the link names that have been assigned to, invoke the Oracle tool
<CRS_HOME>\bin\GuiOracleObjManager.exe
If you already remove the CRS_Home, you can download the tools from:
 - metalink ID: 341214.1
 - http://www.ahmedbaraka.com/oracle/cleanup.zip


3. If you're using RAW disks, invoke the Oracle tool logpatformat.exe to
reinitialize the headers of the disks:
 run logpartformat /q <link name as shown in guioracleobjmanager tool>
 For example: logpartformat /q \\.\ocrcfg
 Repeat this step for all link names listed in guioracleobjmanager)

4. If you're using OCFS:
 navigate to CRS_HOME\bin\
 run logpartformat /q <DRIVELETTER>:
 For example: logpartformat /q P:


5. Remove all the assigned link names using the GUIOracleobjmanager.exe tool
by clicking:
   Placing a check mark in the check box for the given partition, then
choosing 'Commit' from the 'Options' menu.


6. Remove and recreate your logical drives on top of extended partitions from
Windows Disk Management.


7. Use the OUI to remove the software from the CRS home


8. Remove Oracle binaries using Windows explorer, both the CRS home and the
files located in: c:\program files\oracle


9. Check the registry on each node and ensure that the following services have
been removed from (remove them, if not):
 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
 and
 HKEY_LOCAL_MACHINE\SYSTEM\ControlSetXXX\Services

 Services to be removed include:
  ocfs
  OracleCSService
```

```
  OracleEVMService
  OracleCRService
  OracleObjectService
  OracleClusterVolumeService
  OraFenceService


10. Using Windows explorer, remove the following driver files from:
%systemroot%\windows\system32\drivers:
  ocfs.sys
  orafencedrv.sys
  orafenceservice.sys


11. Reboot all servers in your RAC configuration
```

Single Instance to RAC Conversion

The Tools to Convert a Single Instance DB to RAC

- DBCA: enables moving from host that will not be part of the RAC
- rconfig utility: the single instance db host will be part of the RAC
- Enterprise Manager

Conversion Prerequisites for Oracle 10g R2

- Oracle Clusterware 10g Release 2 (10.2) is installed, configured, and running.
- Oracle Real Application Clusters 10g Release 2 (10.2) software is installed.
- The Oracle binary is enabled for RAC.
- Shared storage is available and accessible from all nodes.
- User equivalence exists for the oracle user.

Using rconfig Utitlity

- The example below applies on Oracle 10g R2.

```
Test rconfig before converting to RAC
#make copy of $ORACLE_HOME/assistants/rconfig/sampleXMLs/ConvertToRAC.xml

su - oracle
cd $ORACLE_HOME/assistants/rconfig/sampleXMLs/
cp ConvertToRAC.xml Ora10gToRac.xml

#edit the file: make changes as instructed in the comments
# make sure to set Convert verify = "ONLY"
# below is an example file:

           xsi:schemaLocation="http://www.oracle.com/rconfig">
    <n:ConvertToRAC>
<!-- Verify does a precheck to ensure all pre-requisites are met, before the
conversion is attempted. Allowable values are: YES|NO|ONLY -->
        <n:Convert verify="ONLY">
<!--Specify current OracleHome of non-rac database for SourceDBHome -->

<n:SourceDBHome>/u01/app/oracle/product/10.2.0/db_1</n:SourceDBHome>
<!--Specify OracleHome where the rac database should be configured. It can be
same as SourceDBHome -->

<n:TargetDBHome>/u01/app/oracle/product/10.2.0/db_1</n:TargetDBHome>
<!--Specify SID of non-rac database and credential. User with sysdba role is
required to perform conversion -->
              <n:SourceDBInfo SID="ora10g">
                <n:Credentials>
                  <n:User>sys</n:User>
                  <n:Password>ora10g</n:Password>
                  <n:Role>sysdba</n:Role>
                </n:Credentials>
```

```
              </n:SourceDBInfo>
<!--ASMInfo element is required only if the current non-rac database uses ASM
Storage -->
              <n:ASMInfo SID="+ASM1">
                <n:Credentials>
                  <n:User>sys</n:User>
                  <n:Password>welcome</n:Password>
                  <n:Role>sysdba</n:Role>
                </n:Credentials>
              </n:ASMInfo>
<!--Specify the list of nodes that should have rac instances running.
LocalNode should be the first node in this nodelist. -->
              <n:NodeList>
                <n:Node name="rac1"/>
                <n:Node name="rac2"/>
              </n:NodeList>
<!--Specify prefix for rac instances. It can be same as the instance name for
non-rac database or different. The instance number will be attached to this
prefix. -->
              <n:InstancePrefix>rac</n:InstancePrefix>
<!--Specify port for the listener to be configured for rac database.If
port="", alistener existing on localhost will be used for rac database.The
listener will be extended to all nodes in the nodelist -->
              <n:Listener port=""/>
<!--Specify the type of storage to be used by rac database. Allowable values
are CFS|ASM. The non-rac database should have same storage type. -->
              <n:SharedStorage type="ASM">
<!--Specify Database Area Location to be configured for rac database.If this
field is left empty, current storage will be used for rac database. For CFS,
this field will have directory path. -->
                <n:TargetDatabaseArea>+DG1</n:TargetDatabaseArea>
<!--Specify Flash Recovery Area to be configured for rac database. If this
field is left empty, current recovery area of non-rac database will be
configured for rac database. If current database is not using recovery Area,
the resulting rac database will not have a recovery area. -->
                <n:TargetFlashRecoveryArea>+DG2</n:TargetFlashRecoveryArea>
              </n:SharedStorage>
        </n:Convert>
    </n:ConvertToRAC>
</n:RConfig>


# run the tool for testing
rconfig Ora10gToRac.xml
```

Using DBCA

- The example below tested on an Oracle 10g R2 for Enterprise Linux x86 (version 4.5).
- The DBCA:
  - Automates the configuration of the control file attributes
  - Creates the undo tablespaces and the redo logs
  - Makes the initialization parameter file entries for cluster-enabled environments
  - Configures Oracle Net Services, Oracle Clusterware resources, and the configuration for RAC database management for use by Oracle Enterprise Manager or the SRVCTL utility.

```
# (1) Create a preconfigured image of your single-instance database
$ORACLE_HOME\bin\DBCA

Welcome screen
>Next

Operations screen
>Manage Templates
>Next

Template Management screen
>select "Create a database" template
>"From an existing database (structure as well as data),"
>Next

Source Database screen,
>Select the database name
>Next

On the Template Properties screen
>enter a template name in the Name field.
>enter a description of the file in the Description field
>change the template file location in the Template data file field if you want
>Next

Location of Database Related Files screen
>"Maintain the file locations," so that you can restore the database to the
current directory structure
>Finish
The DBCA generates two files: a database structure file (template_name.dbc)
and a database preconfigured image file (template_name.dfb).


#(2) Copy the preconfigured database image to node1
on rac1 as oracle
scp
oracle@192.168.4.21:/u01/app/oracle/product/10.2.0/db_1/assistants/dbca/templa
tes/Ora10gDB.* /u01/app/oracle/product/10.2.0/db_1/assistants/dbca/templates


#(3) Create the RAC DB from the supplied template
In rac1, run DBCA to create a new database.
On the DBCA Template Selection screen, use the template that you copied.
Complete the installation.

Note: When I tested this procedure, DBCA returns unclear TNS error. I let the
DBCA generate the scritp but unfortunately I then realized that the generated
script will create a single instance database!
```

Administering RAC Database

Administering Oracle Clusterware Components

Managing Cluserware Daemons and Processes

Displaying Clusterware Processes

```
ps -ef | grep d.bin
ps -ef | grep -E "opro|racgimon|evmd|crsd|ocssd" | grep -v grep
```

Starting, Stopping, Enabling and Disabling crs Stack

```
# as root

# prevent CRS from starting on a reboot
crsctl disable crs

# start crs
crsctl start crs

# enable CRS on reboot:
crsctl enable crs

# disable CRS on reboot (not stopping current stack):
crsctl disable crs

# start crs
crsctl stop crs
```

CSS Parameters

- MISSCOUNT: Represents network heartbeat timeouts (600 s)
- DISKTIMEOUT: Represents disk I/O timeouts outside reconfiguration (200 s). Should always be less than I/O latency otherwise crs may trigger node eviction.

```
# to set disktimeout:
1. Shut down Oracle Clusterware on all nodes but one.
2. As root on available node, use (where M is the I/O latency):
crsctl get css disktimeout
crsctl set css disktimeout M+1
3. Reboot available node.
4. Restart all other nodes.

# to set misscount
# same as above except:
crsctl set css misscount 300
```

Administering Voting Disks in RAC

Multiplexing Voting Disks

- Use one voting disk if it is stored on a reliable disk.
- Multiplexed copies should be stored on independent devices.

- Use at least three multiplexed copies.
- A typical voting disk configuration comprises between three and five disks.

Dynamically Adding and Removing Voting Disks after Installing RAC

- Recommendation is to use symbolic links

```
# if you have multiple voting disks, you can add and remove voting disks
su -
crsctl delete css votedisk /dev/raw/raw2
crsctl add css votedisk /dev/raw/raw3

# if the command doesn't work online:
su -
# in all nodes
crsctl stop crs
crsctl add css votedisk /dev/raw/raw3 -force
# in all nodes
crsctl start crs
```

Backing up Voting Disks

- Do it when you add or remove a node.
- Perform it on every voting disk.

```
# Back up the voting disk (can be online)
# usually 4K block size is OK
su -
# to list voting disks currently used
crsctl query css votedisk
# backup
dd if=/dev/raw/raw2 of=~/vdisk.bak bs=4k

# on Windows use ocopy
```

Recovering Voting Disks

```
# recoverying voting disk
dd if=~/vdisk.bak of=/dev/raw/raw2

# if you have multiple voting disks,
#  you can add (multiplex) and remove voting disks
su -
crsctl delete css votedisk /dev/raw/raw2
crsctl add css votedisk /dev/raw/raw2
```

Administering the Oracle Cluster Registry (OCR)

- RAC environments do not support more than two OCRs (OCR and its mirror)
- Using mirrored OCR is optional but recommended when not stored on RAID storage.

Note: The ocrconfig command cannot modify OCR configuration information for nodes that are shut down or for nodes on which the Oracle Clusterware is not running.

Replacing the OCR

- If you receive from ocrcheck the message: "Device/File needs to be synchronized with the other device", it means OCR mirror is out of sync with the primary OCR. In this case, replace or relocate your failing OCR with a copy of the other healthy OCR.
- If it is the primary OCR file that is failing, and if your OCR mirror is still in good health, you can use the ocrconfig –replace ocr <ocrfilename>
- Executing ocrconfig –replace ocr|ocrmirror filename adds the primary or mirror OCR file to your environment if it does not already exist.
- Executing **ocrconfig –replace ocr|ocrmirror** *removes* the primary or the mirror OCR.
- If you remove a primary OCR file, the mirror OCR file becomes primary.

```
# 1) verify the other OCR is online
# 2) verify crs is running in the node you are using to replace OCR
crsctl check crs
crs_stat -t
# 3) run one of the following
ocrconfig -replace ocr /dev/raw/raw5
ocrconfig -replace ocrmirror /dev/raw/raw6
# 4) In any node that is stopped in your RAC
ocrconfig -repair ocrmirror /dev/raw/raw2
```

Adding and Removing the OCR

```
# Adding an Oracle Cluster Registry
ocrconfig -replace ocr /dev/raw/raw5
ocrconfig -replace ocrmirror /dev/raw/raw6

# Removing an Oracle Cluster Registry
# If you remove a primary OCR, then the mirrored OCR becomes the primary OCR
# to remove the ocr
ocrconfig -replace ocr
# to remove the mirrored
ocrconfig -replace ocrmirror
```

Repairing the OCR

- You may need to repair an OCR configuration on a particular node if your OCR configuration changes while that node is stopped.
- The OCR configuration information is stored in:

  /etc/oracle/ocr.loc on Linux and AIX
  /var/opt/oracle/ocr.loc on Solaris and HP-UX
  Registry key HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\ocr on Windows

```
# Reparing an OCR (repairs only the ocr configuration info, not its itegrity)
# 1. the crs must be stopped
su -
crsctl stop crs
# 2. repair
ocrconfig -repair ocrmirror /dev/raw/raw2
# 3. start crs
crsctl start crs
```

Making Physical Backups of the OCR

- Create manually copies of the automatically generated physical backups on daily basis.
- Export the OCR contents before and after making significant configuration changes such as adding or deleting nodes from your environment, modifying Oracle Clusterware resources, or creating a database.

- Do not perform an OCR restore as a correction to revert to previous configurations if some of these configuration changes fail.

```
# backup OCR
# backup the backups that are automatically generated in ( every 4 hrs )
$CRS_HOME/cdata/CLUSTER_NAME
# the default directory can be obtained:
ocrconfig -showbackup

# it is a good idea to set the auto-backup directory to a shared storage:
ocrconfig –backuploc /shared/bak
```

Recovering the OCR using the Physical Backups

```
# make recovery only when you are sure there is an error in the
# ocr. use ocrcheck to check the ocr
# in unix:
# 1. Identify the OCR backups then review the contents of the backup
# backups done timings
ocrconfig -showbackup
ls -lt /u01/crs/cdata/crs/
./ocrdump -backupfile /u01/crs/cdata/crs/backup00.ocr
less OCRDUMPFILE
# 2. Stop the Oracle Clusterware software on ALL Nodes
su -
crsctl stop crs
# 3. Make sure that the OCR devices that you specify in the OCR configuration
file (/etc/oracle/ocr.loc) exist. Restore from an OCR backup file from Step 1:
ocrconfig -restore /u01/crs/cdata/crs/backup00.ocr
# 4. Restart the Oracle Clusterware on all of the nodes
su -
crsctl start crs
# 5. Verify the OCR integrity
cluvfy comp ocr -n all [-verbose]

# in Windows
# 1. Identify the OCR backups using the
ocrconfig -showbackup
ocrdump -backupfile <file_name>
# 2. On all of the remaining nodes, disable the following OCR clients:
OracleClusterVolumeService, OracleCSService, OracleCRService, and the
OracleEVMService
# 3. Apply the OCR backup file from Step 1
ocrconfig -restore c:\oracle\crs\cdata\crs\backup00.ocr
# 4. Start all of the services that were stopped in step 2. Restart all of the
nodes and resume operations in cluster mode.
# 5. Verify the OCR integrity where the -n all
cluvfy comp ocr -n all [-verbose]
```

Making Logical Backups of the OCR (Exporting)

- Export the OCR contents before and after making significant configuration changes, such as adding or deleting nodes from your environment, modifying OracleClusterware resources, or creating a database.

```
# to export OCR contents (output file is a binary not-to-edit file)
ocrconfig –export /u01/ocrbackup/ocr.bak
```

Making Logical Backups of the OCR (Importing)

```
# in Unix
# 1. Identify the OCR export file that you want to import
# 2. Stop Oracle Clusterware on all the nodes in your RAC database
```

```
su -
crsctl stop crs
# 3. Import the file
ocrconfig -import file_name
# 4. Restart Oracle Clusterware on all the nodes
su -
crsctl start crs
# 5. verify OCR integrity
cluvfy comp ocr -n all

# Import in Windows
1. Identify the OCR export file
2. Stop the following OCR clients on each node:
OracleClusterVolumeService, OracleCMService, OracleEVMService,
OracleCSService, and the OracleCRService.
3. Import the OCR export file
ocrconfig -import ocrexport.dat
4. Restart all of the affected services on all nodes.
5. Verify the OCR integrity where node_list is a list of all of the nodes:
cluvfy comp ocr -n all [-verbose]
```

Diagnosing OCR Problems with the OCRDUMP and OCRCHECK Utilities

- OCRDUMP view OCR contents in a readable format.

```
ls -lt /u01/crs/cdata/crs/
./ocrdump -backupfile /u01/crs/cdata/crs/backup00.ocr
less OCRDUMPFILE

./ocrdump dumpoutput -backupfile /u01/crs/cdata/crs/backup00.ocr
./ocrdump -stdout -backupfile /u01/crs/cdata/crs/backup00.ocr -xml

# checks logs are reported in CRS_Home/log/hostname/client
./ocrcheck
```

Administering Storage

Datafile Access in Real Application Clusters

Redo Log File Storage in Real Application Clusters

- Each instance has its own online redo log groups which are referred to as an instance's thread of online redo.

```
/* instance Thread */
-- thread is unique to every node
select value from v$parameter where name='thread';

-- If you change the thread of an instance, it won't
--  take effect till you enable it (otherwise db restart will fail):
ALTER DATABASE ENABLE THREAD 3;

-- following commands remove thread 3 and replace it with thread 2
ALTER SYSTEM SET thread = 2 SCOPE=SPFILE SID='rac2';
/u01/crs11g/bin/srvctl stop instance -d rac -i rac2
/u01/crs11g/bin/srvctl start instance -d rac -i rac2
connect / as sysdba
alter database disable thread 3;

-- every thread has at least two groups
select thread#, group#, members, bytes/1024/1024 MB, status from v$log order
by thread#, group#;
select group#, member, status from v$logfile order by group#;


/* to add a redo log group */
-- if OMF or ASM is used
ALTER DATABASE ADD LOGFILE THREAD 1 GROUP 5 SIZE 50M;
ALTER DATABASE ADD LOGFILE THREAD 2 GROUP 6 SIZE 50M;
-- or
alter database add logfile thread 2
group 5 ('/../_g5_m1.dbf', '/../_g5_m2.dbf') SIZE 128M REUSE,
group 6 ('/../_g6_m1.dbf', '/../_g6_m2.dbf') SIZE 128M REUSE,
group 7 ('/../_g7_m1.dbf', '/../_g7_m2.dbf') SIZE 128M REUSE,
group 8 ('/../_g8_m1.dbf', '/../_g8_m2.dbf') SIZE 128M REUSE;

/* to drop  a group */
-- make sure its inactive
select thread#, group#, members, bytes/1024/1024 MB, status from v$log order
by thread#, group#;
-- switch log if required
alter system switch loglife;
-- start archiving, if required: specific instance
alter system archive log instance 'rac1' next ;
-- drop the group
ALTER DATABASE DROP LOGFILE GROUP 5;
ALTER DATABASE DROP LOGFILE GROUP 6;
```

Automatic Undo Management in Real Application Clusters

```
# display undo tablespace used by the SID
SELECT VALUE FROM V$PARAMETER WHERE UPPER(NAME) in
```

```
('UNDO_TABLESPACE','INSTANCE_NAME');

# change undo tablespace for an instance
ALTER SYSTEM SET UNDO_TABLESPACE ='UNDORAC2' SID='rac2';
```

Administering ASM Instances with SRVCTL in RAC

```
# to start, stop, and obtain the status of an ASM instance
srvctl start asm -n node_name [-i asm_instnace_name] [-o start_options] [-c
<connect_str> | -q]
srvctl stop asm -n node_name [-i asm_instance_name] [-o stop_options] [-c
<connect_str> | -q]
srvctl config asm -n node_name
srvctl status asm -n node_name

# to add configuration information about an existing ASM instance:
srvctl add asm -n node_name -i asm_instance_name -o oracle_home

# to remove an ASM instance:
srvctl remove asm -n node_name [-i asm_instance_name]

# to enable an ASM instance:
srvctl enable asm -n node_name [-i ] asm_instance_name

# to disable an ASM instance:
srvctl disable asm -n node_name [-i asm_instance_name]
```

Administering Cluster Databases

Displaying Current Instance in SQL*Plus Prompt

```
# in login.sql file (usually in $ORACLE_HOME/bin) add:
SET SQLPROMPT '_USER@ _CONNECT_IDENTIFIER> '
```

Starting and Stopping Instances and RAC Databases

Starting Up and Shutting Down with SQL*Plus

```
-- on an instance on your local node
echo $ORACLE_SID
CONNECT / AS SYSDBA
SELECT VALUE FROM V$PARAMETER WHERE UPPER(NAME) ='INSTANCE_NAME';
-- the following command will wait for all trans in the db
SHUTDOWN TRANSACTIONAL
-- the following command will wait for the trans in the connected node only
SHUTDOWN TRANSACTIONAL LOCAL
-- connected instance will shutdown
STARTUP MOUNT

-- or
CONNECT /@db1 AS SYSDBA
..
```

Intermittent Windows Shutdown Issue in RAC Environments

Normal Windows shutdown may hange, if the instance was shutdown using SQL*Plus. To avoid it, shutdown the following using SRVCTL:

- All services on the node.
- All database instances on the node.
- All ASM instances on the node.
- All node applications on the node.

Starting Up and Shutting Down with SRVCTL

- Shutting down a database means shutting down all its instances.

```
# specific instance(s)
# preferred and available services will also be started alongside
srvctl start instance -d db_name -i "inst_name_list" [-o start_options]
srvctl stop instance -d name -i "inst_name_list" [-o stop_options]
srvctl stop instance -d rac -i "rac1,rac2" -o immediate

# entire cluster database
# when you start, only non-running instances will be started
srvctl start database -d name [-o start_options]
srvctl start database -d rac -o mount
srvctl stop database -d name [-o stop_options]
```

Customizing How Oracle Clusterware Manages RAC Databases

You may want to:

- prevent the Oracle Clusterware from restarting your RAC database when you restart the system
- avoid restarting failed instances more than once

```
# when you make a service unavailable, it won't run under the Oracle crs
#  for automatic startup, failover, or restart
srvctl disable database -d rac
srvctl disable instance -d rac -i "rac1,rac2"
srvctl disable service -d rac -s hrserv,marketing
# -n node name
srvctl disable asm -n crmnode1 -i asm1
```

Switching Between the Database Automatic and Manual Policies

- When AUTO_START attribute is set to 2 (MANUAL) for a resource, crs will not automatically start it on reboot.

```
# to display the current policy
srvctl config database -d rac -a
..
POLICY:  AUTOMATIC

# change the current policy to another one
srvctl modify database d database_name -y AUTOMATIC|MANUAL
```

Customizing Resource Parameters (like AUTO_START)

- You can customize the following resource parameters for database or ASM instances, databases, services, and service members:

  # when 1 (0), it atuo-restarts on system reboot

  AUTO_START as

  # restart attempts before relocate (1)

  RESTART_ATTEMPTS ra

  # application up-time to be considered stable by crs

  UPTIME_THRESHOLD ut

```
# 1. retreive resource name
crs_stat –t

# 2. update the OCR with the right attribute values for your resources
crs_register ora.….inst -update -o as=2,ra=1,ut=7d
crs_register ora.….asm -update -o as=2,ra=1,ut=7d
crs_register ora.….db -update -o as=2,ra=1,ut=7d
crs_register ora.….cs -update -o as=2,ra=0
crs_register ora.….svr -update -o as=2,ra=0
```

Handling Initialization Parameter Files in RAC

Setting Server Parameter File Parameter Values for Real Application Clusters

- Practically, each instance uses its own PFILE file that contains only one parameter called SPFILE. The SPFILE parameter points to the shared SPFILE on your shared storage.

```
/* about the parameters in spfile in RAC */
# to know wheather spfile or pfile was used when startup
SELECT DECODE(value, NULL, 'PFILE', 'SPFILE:'|| value) "Init File Type"
FROM sys.v_$parameter WHERE name = 'spfile';

-- spfile is shared by all instances
cat $ORACLE_HOME/dbs/initrac1.ora
SPFILE='+DG1/rac/spfilerac.ora'

-- example
ALTER SYSTEM SET OPEN_CURSORS=500
        COMMENT='Changed by Ahmed for testing on 16-dec-09'
        SCOPE=BOTH
        SID='*';
ALTER SYSTEM SET OPEN_CURSORS=1000 SPFILE=BOTH SID='rac1';

-- to remove a paramter from spifle
ALTER SYSTEM RESET timed_statistics SCOPE=SPFILE SID='*';
ALTER SYSTEM SET timed_statistics = '' SCOPE=SPFILE;
```

Parameters Used in RAC Databases

CLUSTER_DATABASE must be TRUE

CLUSTER_DATABASE_INSTANCES

CLUSTER_INTERCONNECTS

DB_NAME

DISPATCHERS

INSTANCE_NAME

SERVICE_NAMES

SESSIONS_PER_USER If SESSIONS_PER_USER is set to 1 for a user, the user can log on to the database more than once as long as each connection is from a different instance.

THREAD

SPFILE

MAX_COMMIT_PROPAGATION_DELAY deprecated in 10g R2

Parameters that Must Have Identical Settings on All Instances

ACTIVE_INSTANCE_COUNT

ARCHIVE_LAG_TARGET

CLUSTER_DATABASE

CLUSTER_DATABASE_INSTANCES number of instances in the RAC

CONTROL_FILES

DB_BLOCK_SIZE

DB_DOMAIN

DB_FILES

DB_NAME

DB_RECOVERY_FILE_DEST

DB_RECOVERY_FILE_DEST_SIZE

DB_UNIQUE_NAME

UNDO_MANAGEMENT

Parameters That Must Have Unique Settings on All Instances

THREAD

ROLLBACK_SEGMENTS

UNDO_TABLESPACE if automatic is used

Parameters that Should Have Identical Settings on All Instances

It is highly recommended to set same value for parameters in the following list in all the instances:

ARCHIVE_LAG_TARGET

LICENSE_MAX_USERS

LOG_ARCHIVE_FORMAT

SPFILE

TRACE_ENABLED

UNDO_RETENTION

ASM Instance Initialization Parameters and RAC

CLUSTER_DATABASE must be TRUE

ASM_DISKSTRING Multiple instances can have different values (not recommended)

ASM_POWER_LIMIT Multiple instances can have different values

```
export ORACLE_SID=+ASM1

sqlplus /nolog
conn / as sysdba

select name, value
 from v$parameter
 where upper(name) in ('CLUSTER_DATABASE','ASM_DISKSTRING','ASM_POWER_LIMIT');
```

Dropping a RAC Database

- Option 1: After making sure the crs and the database are up and running, issue dbca and delete the database.
- Option 2: issue the commands below:

```
startup mount exclusive restrict;
drop database;
# to delete archived log and the backups as well
drop database including backups;

srvctl remove instance -d rac -i rac1
srvctl remove database -d rac
```

Workload Management in RAC

Types of Workload Distribution

- Connection load balancing:
  - Client-side connect-time load balancing
  - Client-side connect-time failover
  - Server-side connect-time load balancing
- Run-time connection load balancing can be implemented by using connection pools in middle tier.

Connection Load Balancing

Client-Side Load Balancing and Failover

- If you use connect-time failover, do not set GLOBAL_DBNAME in the listener.ora

```
#  to enable Client-Side Load Balancing ONLY (i.e., no failover): send request
#   randomly to one listener from the list.
CSLB_ONLY =
    (DESCRIPTION =
        (LOAD_BALANCE = ON)
        (ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip)(PORT = 1521))
        (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip)(PORT = 1521))
        (CONNECT_DATA =
            (SERVER = DEDICATED)
            (SERVICE_NAME = racdb)
        )
    )

# use this settings for RAC database
# to enable failover (using TCP/IP time-out, the client make sure the host is
#  alive before sending connection requrest) (in RAC, you use vip addresses)
CSLB_ONLY =
    (DESCRIPTION =
        (LOAD_BALANCE = ON)
        (FAILOVER = ON)
        (ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip)(PORT = 1521))
        (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip)(PORT = 1521))
        (CONNECT_DATA =
            (SERVER = DEDICATED)
            (SERVICE_NAME = racdb)
        )
    )
```

Server-Side Load Balancing

- Target: which listener should take the connection.

```
-- Configuring Server-side connection load balancing
-- 1) define the service which will be used for the connection load balancing
-- 2) Add entries in every client's TNSNAMES.ORA file for the new alias
```

```
HRSERV =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip.mydomain.com)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip.mydomain.com)(PORT = 1521))
    )
   (LOAD_BALANCE = ON) -- not related to server-side load balancing
   (FAILOVER = ON)     -- not related to server-side load balancing
    (CONNECT_DATA =
      (SERVICE_NAME = hrserv)
    )
  )
-- 3)  Add entries in the TNSNAMES.ORA file of every node to
--      include the REMOTE_LISTENER setting
LISTENERS_RAC =
    (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = rac1-vip.mydomain.com)(PORT =
1521))
        (ADDRESS = (PROTOCOL = TCP)(HOST = rac2-vip.mydomain.com)(PORT =
1521))
    )
--  4) set the parameter to make PMON automatically register the database
--      with all the listeners in the nodes
ALTER SYSTEM SET REMOTE_LISTENER =LISTENERS_RAC SID='*' SCOPE=BOTH;


-- setting the default connection load balancing goal for the service hrserv
--  to LONG (long-lived apps like forms)(default), SHORT (short-lived apps)
execute dbms_service.modify_service (service_name => 'hrserv' , clb_goal =>
dbms_service.clb_goal_long);
execute dbms_service.modify_service (service_name => 'hrserv' , clb_goal =>
dbms_service.clb_goal_short);

-- view it (name is case sensitive)
SELECT CLB_GOAL FROM DBA_SERVICES WHERE NAME='hrserv';
```

Fast Application Notification (FAN)

- FAN publishing load balancing advisory events and UP and DOWN info about services, instances and nodes to client applicaitons.
- You can take advantage of FAN events in the following three ways:
    1. Application can take advantage of FAN without any programmatic changes by utilizing an integrated Oracle Client like: Oracle Database 10g JDBC, ODP.NET, and OCI.
    2. Applications can take advantage of FAN programmatically by using the Oracle Notification Service (ONS) Application Programming Interface (API) to subscribe to FAN events and execute event handling actions upon the receipt of an event.
    3. You can implement FAN with server side callouts on your database tier.

Using Fast Application Notification Callouts

- FAN callouts are server side executables that RAC executes immediately when high availability events occur.

```
# place an executable in the directory CRS_home/racg/usrco in all nodes
vi /u01/crs/racg/usrco/callout.sh
#! /bin/ksh
FAN_LOGFILE=/home/oracle/log/rac_`hostname`.log
echo $* "reported="`date` >> $FAN_LOGFILE &
```

```
# place an executable in the directory CRS_home/racg/usrco in all nodes
# example source is Oracle Documentation 10g R1
vi /u01/crs/racg/usrco/callout.sh
#!/usr/bin/sh#

# Description: wrapper script to enable RAC event logging and notification
#          to generic third-party systems.  The script showcases two possible
#          methods to enable local or remote logging/notification of RAC-
#          detected events.#

AWK=/usr/bin/awk
MY_CRS_HOME=/private/oracle/crs

# Scan and parse arglist:#
for ARGS in $*; do
   PROPERTY=`echo $ARGS | $AWK -F"=" '{print $1}'`
   VALUE=`echo $ARGS    | $AWK -F"=" '{print $2}'`

   #> map EVTTYPE to EVENT_TYP, NODE to HOST:
   case $PROPERTY in
     #> EVENT_TYP is one of: NODE, DATABASE, INSTANCE, SERVICE, SERVICEMEMBER
     EVENT_TYP | event_typ)     NOTIFY_EVENT_TYP=$VALUE ;;
     VERSION | version)         NOTIFY_VERSION=$VALUE ;;
     SERVICE | service)         NOTIFY_SERVICE=$VALUE ;;
     DATABASE | database)       NOTIFY_DBNAME=$VALUE ;;
     INSTANCE | instance)       NOTIFY_INSTANCE=$VALUE ;;
     HOST | host)               NOTIFY_HOST=$VALUE ;;
     STATUS | status)           NOTIFY_STATUS=$VALUE ;;
     TIMESTAMP | timestamp)     NOTIFY_SVRLOGDATE=$VALUE ;;
   esac
done


# ######################################################
# [1] Notification Method 1:  On-cluster file logging
# ######################################################
# This section simply writes one-line entries for each event published by RAC,
# and the log is written to standard RAC log directory.  It will blindly
# record all RAC events, regardless of state (UP, DOWN or NOT_RESTARTING):

RACEVT_LOGFILE=$MY_CRS_HOME/racg/log/rac_${NOTIFY_SERVICE}_uptime.log

echo RAC\(v$NOTIFY_VERSION\):  $NOTIFY_STATUS event, type "$NOTIFY_EVENT_TYP",
\
     `if [ -n "$NOTIFY_SERVICE" ]; then \
        echo "for service $NOTIFY_SERVICE"
      fi` \
     \[`if [ -n "$NOTIFY_INSTANCE" ]; then \
        echo "inst: $NOTIFY_INSTANCE"
        fi` \
       `if [ -n "$NOTIFY_DATABASE" ]; then \
        echo "db: $NOTIFY_DATABASE"
        fi` \
       `if [ -n "$NOTIFY_HOST" ]; then \
        echo "db: $NOTIFY_HOST"
        fi` \
```

```
     \] received on $NOTIFY_SVRLOGDATE >>  $RACEVT_LOGFILE



# ###########################################################
# [2] Notification Method 2:  On-cluster program execution
# ###########################################################
# Let's assume you have a custom client program in /tmp (say logTicket)
# to which you can pass certain arguments.  This program connects to a
# customer-service application that processes incident tickets for your
# IT department:
#    % /tmp/logTicket {serverside_timestamp} \
#                     {databasename} {servicename} \
#                     {instancename} {hostname}#

# Let us also assume that a ticket would be logged only for NOT_RESTARTING
# events, as they are the ones that exceeded RAC-monitored timeouts and
# seriously need human intervention for full resolution.#

# -------------------
# ONE SOLUTION TO [2]:
# -------------------
if [ $NOTIFY_STATUS  = "NOT_RESTARTING" -o $NOTIFY_STATUS = "not_restarting"
]; then
  /tmp/logTicket $NOTIFY_SVRLOGDATE $NOTIFY_DBNAME \
                        $NOTIFY_SERVICE \
                        $NOTIFY_INSTANCE $NOTIFY_HOST   >> $RACEVT_LOGFILE
fi
```

Configuring the Server-Side ONS

- When useocr=on in ons.config, racgons add_config command store this data in OCR.

```
# target: to add rac3 node in the RAC ONS
# (1) obtain the remote nodes from rac1,rac2
#  and set useocr=on
cat $CRS_HOME/opmn/conf/ons.config

# (2) in rac3:
racgons add_config rac1:6200 rac2:6200
racgons add_config rac3:6200

# (3) in rac1, rac2:
onsctl reconfig
```

Administering Load Balancing Advisory

- The load balancing advisory has the task of advising the direction of incoming work to the RAC instances that provide optimal quality of service for that work.
- To test the load balancing, you can use the scripts in the appendix to apply load and the following section for monitoring.

```
-- view load balancing gaol of a current service
-- NONE means load balancing advisory is disabled
SELECT NAME, GOAL, CLB_GOAL FROM DBA_SERVICES;

-- SERVICE TIME: response-time based, like online shopping
```

```
execute dbms_service.modify_service (service_name => 'hrserv' -
  , goal => dbms_service.goal_service_time -
  , clb_goal => dbms_service.clb_goal_short);
-- THROUGHPUT: rat at which jobs are completed, like batch processing
execute dbms_service.modify_service (service_name => 'hrserv' -
 , goal => dbms_service.goal_throughput -
 , clb_goal => dbms_service.clb_goal_long);
-- NONE: LBA disabled
execute dbms_service.modify_service (service_name => 'hrserv' -
 , goal => dbms_service.goal_none -
 , clb_goal => dbms_service.clb_goal_long);
```

Monitoring Load Balancing Advisory

- V$SERVICEMETRIC : service metrics on the **most** 5-second and 1-minute intervals
- V$SERVICEMETRIC_HISTORY : recent **history** of the metric values

```
SELECT
     service_name
    ,TO_CHAR(begin_time,'hh24:mi:ss') beg_hist
    ,TO_CHAR(end_time,'hh24:mi:ss') end_hist
    ,inst_id
    ,goodness
    ,delta
    ,flags
    ,cpupercall
    ,dbtimepercall
    ,callspersec
    ,dbtimepersec
  FROM gv$servicemetric
 WHERE service_name ='hrserv'
 ORDER BY service_name, begin_time DESC, inst_id;

SELECT SERVICE_NAME,
 to_char(BEGIN_TIME,'hh24:mi:ss') BEGIN_TIME,
 to_char(END_TIME,'hh24:mi:ss') END_TIME,
 INTSIZE_CSEC, GROUP_ID "Service Metric Group",
 CPUPERCALL "CPU time per call",
 DBTIMEPERCALL "Elapsed time per call",
 CALLSPERSEC "User Calls/s",
 DBTIMEPERSEC "DB Time/s"
from V$SERVICEMETRIC_HISTORY
order by SERVICE_NAME, BEGIN_TIME desc;

-- aggregated
SELECT
 SERVICE_NAME,
 GROUP_ID "Service Metric Group",
 round(sum(CPUPERCALL),2) "CPU time per call",
 round(sum(DBTIMEPERCALL),2) "Elapsed time per call",
 round(sum(CALLSPERSEC),2) "User Calls/s",
 round(sum(DBTIMEPERSEC),2) "DB Time/s"
from V$SERVICEMETRIC_HISTORY
group by  SERVICE_NAME, GROUP_ID
order by SERVICE_NAME;
```

Transparent Application Failover (TAF)

TAF Basic Configuration without FAN (From Client Side)

- FAILOVER_MODE must be configured in tnsnames.ora
- If using TAF, do not set the GLOBAL_DBNAME parameter in the listener.ora

```
# when you create the service
srvctl add service -d rac -s hrserv -r rac1,rac2 -P BASIC
srvctl start service -d rac -s hrserv

# in the client tnsnames.ora
# TYPE: SELECT, SESSION
hrserv =
(DESCRIPTION =(FAILOVER=ON)(LOAD_BALANCE=ON)
 (ADDRESS=(PROTOCOL=TCP)(HOST=rac1-VIP)(PORT=1521))
 (ADDRESS=(PROTOCOL=TCP)(HOST=rac2-VIP)(PORT=1521))
  (CONNECT_DATA =
   (SERVICE_NAME = hrserv)
   (FAILOVER_MODE =
    (TYPE=SESSION)
    (METHOD=BASIC)
    (RETRIES=100)
    (DELAY=5))))
```

TAF Basic Configuration with FAN (Server-Side)

- You not need to specify TAF parameters in the tnsnames.ora

```
srvctl add service -d rac -s hrserv -r rac1,rac2

srvctl start service -d RACDB -s AP

exec dbms_service.modify_service ( ,-
 service_name => 'hrserv' ,-
 aq_ha_notifications => true ,-
 failover_method => dbms_service.failover_method_basic ,-
 failover_type => dbms_service.failover_type_session ,-
 failover_retries => 100, failover_delay => 5 ,-
 clb_goal => dbms_service.clb_goal_long);

hrserv =
(DESCRIPTION =(FAILOVER=ON)(LOAD_BALANCE=ON)
 (ADDRESS=(PROTOCOL=TCP)(HOST=rac1-VIP)(PORT=1521))
 (ADDRESS=(PROTOCOL=TCP)(HOST=rac2-VIP)(PORT=1521))
 (CONNECT_DATA = (SERVICE_NAME = hrserv)))
```

TAF Preconnect Configuration

- Fast failover but more resources.
- The shadow service is always named using the format <service_name>_PRECONNECT.

```
# configure the preferred and available nodes
srvctl add service -d rac -s hrserv -r rac1 -a rac2 -P PRECONNECT
srvctl start service -d rac -s hrserv

HRSERV =
(DESCRIPTION =(FAILOVER=ON)(LOAD_BALANCE=ON)
```

```
 (ADDRESS=(PROTOCOL=TCP)(HOST=rac1-vip)(PORT=1521))
 (ADDRESS=(PROTOCOL=TCP)(HOST=rac2-vip)(PORT=1521))
  (CONNECT_DATA = (SERVICE_NAME = hrserv)
   (FAILOVER_MODE = (BACKUP=HRSERV_PRECONNECT)
    (TYPE=SESSION)(METHOD=PRECONNECT))))

HRSERV_PRECONNECT =
(DESCRIPTION =(FAILOVER=ON)(LOAD_BALANCE=ON)
 (ADDRESS=(PROTOCOL=TCP)(HOST=rac1-vip)(PORT=1521))
 (ADDRESS=(PROTOCOL=TCP)(HOST=rac2-vip)(PORT=1521))
  (CONNECT_DATA = (SERVICE_NAME = HRSERV_PRECONNECT)))
```

Verifying TAF Configuration

```
SELECT machine, failover_method, failover_type,
failed_over, service_name, COUNT(*)
FROM v$session
GROUP BY machine, failover_method, failover_type,
failed_over, service_name;

select instance_name from v$instance;
```

Enabling Distributed Transaction Processing for Services

```
# For services that you are going to use for distributed transaction
processing,
# define only one instance as the preferred instance
srvctl add service -d crm -s xa_01.service.us.oracle.com -r RAC01 -a RAC02,
RAC03
# mark the service for distributed trans processing by setting DTP to TRUE:
execute dbms_service.modify_service(service_name
=>'xa_01.mydomain.com',DTP=>TRUE);
```

Administering Services

Service Attributes

- Global unique name
- Network name
- Load Balancing Advisory goal: best service quality (service response time), or best throughput (how much work is completed in a unit of time).
- Distributed transactions flag
- Advance queuing notification characteristics for OCI and ODP.NET clients
- Failover characteristics
- Connection load-balancing algorithm: SHORT: Use Load Balancing Advisory, LONG: Using session count by service
- Threshold: for response time and CPU consumption
- Priority: services to consumer groups mapping
- High-availability configuration: how the service is distributed across instances when the system first starts

Administering Services with DBCA

```
1. On the DBCA Welcome page, select the Oracle Real Application Clusters
option ->Next
2. On the DBCA Operations: select Services Management ->Next
3. On the DBCA List of Databases: select the cluster database ->Next

-- confirm
select value from v$parameter where upper(name)='SERVICE_NAMES'
```

Administering Services with PL/SQL

```
/* Create a Service (not a recommended method in RAC)*/
-- some attributes can only be modified using PL/SQL
-- service_name and network_name are mandatory
-- others are optional will NULL defaults
begin
DBMS_SERVICE.CREATE_SERVICE(
  service_name =>'olapserv',
  network_name =>'olapserv', -- TNS name for connections to the service
  goal =>DBMS_SERVICE.GOAL_SERVICE_TIME, -- or GOAL_THROUGHPUT, GOAL_NONE
  dtp  =>FALSE, -- is it for distributed transaction
  aq_ha_notifications   =>TRUE, -- should HA events sent to AQ
  failover_method =>DBMS_SERVICE.FAILOVER_METHOD_BASIC, -- or _NONE
  failover_type =>DBMS_SERVICE.FAILOVER_TYPE_SESSION, -- or _NONE _SELECT
  failover_retries =>10,
  failover_delay=>1 , -- in seconds
  clb_goal=>DBMS_SERVICE.CLB_GOAL_SHORT -- or CLB_GOAL_LONG
   );
end;
```

```
/

select name from dba_services ;

/* Modify a Service Charactaristic */
-- some attributes can only be modified using PL/SQL
-- if stopped, start it before modify
begin
DBMS_SERVICE.MODIFY_SERVICE(
  service_name =>'olapserv',
  goal =>DBMS_SERVICE.GOAL_SERVICE_TIME, -- or GOAL_THROUGHPUT, GOAL_NONE
  dtp  =>FALSE, -- is it for distributed transaction
  aq_ha_notifications   =>TRUE, -- should HA events sent to AQ
  failover_method =>DBMS_SERVICE.FAILOVER_METHOD_BASIC, -- or _NONE
  failover_type =>DBMS_SERVICE.FAILOVER_TYPE_SESSION, -- or _NONE _SELECT
  failover_retries =>10,
  failover_delay=>1 , -- in seconds
  clb_goal=>DBMS_SERVICE.CLB_GOAL_SHORT -- or CLB_GOAL_LONG
   );
end;
/


/* Delete Service */
begin
 DBMS_SERVICE.DELETE_SERVICE( service_name =>'olapserv');
end;
/


/* Start a service */
begin
DBMS_SERVICE.START_SERVICE(service_name=>'hrserv',
 instance_name => DBMS_SERVICE.ALL_INSTANCES, -- if NULL current inst
 );
end;
/


/* Stop Service */
begin
DBMS_SERVICE.STOP_SERVICE(service_name=>'hrserv',
 instance_name => DBMS_SERVICE.ALL_INSTANCES, -- if NULL current inst
 );
end;
/


/* Disconnect Sessions */
-- sessions connected to the service are terminated
begin
 -- CAUTION: control doesn't return till all sessions are terminated
 DBMS_SERVICE.DISCONNECT_SESSION('hrserv');
end;
/
```

Administering Services with SRVCTL

- Creating a Service

```
srvctl add service -d dbname -s service_name -r preferred_list [-a
available_list] [-P TAF_policy]
```

```
 taf policy: NONE, BASIC, or PRECONNECT
```

```
/* Creating Services */
# it does NOT add the required tnsnames.ora settings
srvctl add service –d rac –s GL -r RAC02 -a RAC01
srvctl add service –d rac –s ERP –r RAC01,RAC02 -a RAC03,RAC04

/* Starting and Stopping Services */
srvctl start service -d dbname [-s service_name_list] [-i inst_name] [-o
start_options]
srvctl stop service -d dbname [-s service_name_list [-i inst_name]] [-c
connect_str]

/* Enabling and Disabling Services */
srvctl enable service -d dbname -s service_name_list [-i inst_name]
srvctl disable service -d dbname -s service_name_list [-i inst_name]

/* Relocating Services */
srvctl relocate service -d rac -s hrserv -i rac2 -t rac3

/* upgrading an available instance as a preferred instance for a Service*/
-- see also next section
srvctl modify service –d rac –s hrserv -i rac3 –r

/* Obtaining the Statuses of Services */
srvctl status service -d rac -s hrserv

/* Obtaining the Configuration of Services */
srvctl config service -d rac -s hrserv -a

/* Removing Services */
rvctl stop service -d rac -s hrserv
rvctl remove service -d rac -s hrserv
```

Controlling the Preferred and Available Instances

```
/* Upgrading an Available instance as a Preferred instance for a Service */
-- -r upgrade instance to preferred
srvctl modify service –d rac –s hrserv -i rac3 –r


/* the scenario below makes:
    - i1-i4 preferred, i5-i6 available for dwserv
    - i5-i6 preferred, i1-i4 available for hrserv
*/
-- changes take effect on next services restart
-- if used, -f Disconnect all sessions during stop
srvctl modify service –d PROD –s dwserv –n –i I1,I2,I3,I4 –a I5,I6
srvctl modify service –d PROD –s hrserv –n –i I5,I6 –a I1,I2,I3,I4

-- –f stops the services globally on your cluster
```

```
srvctl stop service –d rac –s dwserv,hrserv -f
srvctl start service –d rac –s dwserv,hrserv
```

Using Services with Client Applications

```
/* TNS connection descriptor */
-- LOAD_BALANCE=on only enables a client-side connection load balancing
ERP=(DESCRIPTION=
(LOAD_BALANCE=on)
(ADDRESS=(PROTOCOL=TCP)(HOST=node-1vip)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=node-2vip)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=node-3vip)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=node-4vip)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=ERP)))

/* JDBC connection description (1) */
url="jdbc:oracle:oci:@ERP"

/* JDBC connection description (2) */
url="jdbc:oracle:thin:@(DESCRIPTION=
(LOAD_BALANCE=on)
(ADDRESS=(PROTOCOL=TCP)(HOST=node-1vip)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=node-2vip)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=node-3vip)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=node-4vip)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=ERP)))"
```

Services and the Scheduler

```
BEGIN
 DBMS_SCHEDULER.CREATE_JOB_CLASS (
  job_class_name => 'my_jobs_class',
  SERVICE => 'hrserv');
END;
/

BEGIN
dbms_scheduler.set_attribute(
 name => 'my_jobs_class',
 attribute => 'service',
 value => 'hrserv');
END;
/

begin
 dbms_scheduler.create_job (
 job_name => 'My_Job',
 job_type => 'plsql_block',  -- stored_procedure, plsql_block, executable,
chain
 job_action => 'begin null; end;',
 job_class  => 'my_jobs_class',
 start_date => SYSDATE+1,
 repeat_interval => 'FREQ=MINUTELY; INTERVAL=2');
end;
```

```
/

BEGIN
dbms_scheduler.set_attribute(
 name => 'My_Job',
 attribute => 'INSTANCE_STICKINESS',
 value => TRUE);
END;
/

BEGIN
 DBMS_SCHEDULER.DROP_JOB('My_Job');
END;
/

BEGIN
 DBMS_SCHEDULER.DROP_JOB_CLASS('my_jobs_class');
END;
/
```

```
DBMS_SCHEDULER.CREATE_JOB_CLASS(
JOB_CLASS_NAME => 'HOT_BATCH_CLASS',
RESOURCE_CONSUMER_GROUP => NULL ,
SERVICE => 'HOT_BATCH_SERV' ,
LOGGING_LEVEL => DBMS_SCHEDULER.LOGGING_RUNS,
LOG_HISTORY => 30, COMMENTS => 'P1 batch');

DBMS_SCHEDULER.CREATE_JOB(
JOB_NAME => 'my_report_job',
JOB_TYPE => 'stored_procedure',
JOB_ACTION => 'my_name.my_proc();',
NUMBER_OF_ARGUMENTS => 4, START_DATE => SYSDATE+1,
REPEAT_INTERVAL => 5, END_DATE => SYSDATE+30,
JOB_CLASS => 'HOT_BATCH_CLASS',
ENABLED => TRUE,
AUTO_DROP => false, COMMENTS => 'daily status');
```

Measuring Performance by Service Using the AWR

```
/* Enabling Gather statiscitc on Service-Module-Action combination */
-- to display currently connected services, modules and actions
select SID, USERNAME, SERVICE_NAME, MODULE, ACTION from V$SESSION
where SERVICE_NAME in ('hrserv','oeserv')

-- service name and module name are mandatory
begin
 DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(SERVICE_NAME => 'hrserv',
   MODULE_NAME=>'PAYROLL',
   ACTION_NAME => 'EXCEPTIONS PAY');
end;
/

-- gather stats for PAYROLL module and ACTION whose name is null
begin
 DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(SERVICE_NAME => 'hrserv',
```

```
  MODULE_NAME=>'PAYROLL',
  ACTION_NAME => NULL);
end;
/

-- gather stats for PAYROLL module and All its ACTIONs
begin
 DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(SERVICE_NAME => 'hrserv',
  MODULE_NAME=>'PAYROLL',
  ACTION_NAME => '###ALL_ACTIONS');
end;
/

-- to view enabled monitorings
-- types: SERVICE, SERVICE_MODULE, SERVICE_MODULE_ACTION
select A.AGGREGATION_TYPE, A.PRIMARY_ID , A.QUALIFIER_ID1 , A.QUALIFIER_ID2
from DBA_ENABLED_AGGREGATIONS a

-- to view gathered stats
select S.AGGREGATION_TYPE, S.SERVICE_NAME, S.MODULE, S.ACTION, N.CLASS,
       decode(n.CLASS,
'1','User','2','Redo','4','Enqueue','8','Cache','16','OS','32','RAC','64','SQL
','128','Debug', N.CLASS) STAT_CLASS,
       S.STAT_NAME, S.VALUE
from   V$SERV_MOD_ACT_STATS s, V$STATNAME n
where  S.STAT_ID = N.STAT_ID
order by N.CLASS, S.STAT_ID

-- call times and performance statistics views:
V$SERVICE_STATS
V$SERVICE_EVENTS
V$SERVICE_WAIT_CLASSES
V$SERVICEMETRIC
V$SERVICEMETRIC_HISTORY

/* To Disable Cumulative Stats */
-- stats will be removed from V$SERV_MOD_ACT_STATS
begin
 DBMS_MONITOR.SERV_MOD_ACT_STAT_DISABLE(SERVICE_NAME => 'hrserv',
   MODULE_NAME=>'PAYROLL',
   ACTION_NAME => 'EXCEPTIONS PAY');
end;
/


/* Service Quality Statistics */
-- script from Oracle documentation
-- provides service quality statistics every five seconds
SET PAGESIZE 60 COLSEP '|' NUMWIDTH 8 LINESIZE 132 VERIFY OFF FEEDBACK OFF
COLUMN service_name FORMAT A20 TRUNCATED HEADING 'Service'
COLUMN begin_time HEADING 'Begin Time' FORMAT A10
COLUMN end_time HEADING 'End Time' FORMAT A10
COLUMN instance_name HEADING 'Instance' FORMAT A10
COLUMN service_time HEADING 'Service Time|mSec/Call' FORMAT 999999999
COLUMN throughput HEADING 'Calls/sec'FORMAT 99.99
BREAK ON service_name SKIP 1
SELECT
```

```
service_name
, TO_CHAR(begin_time, 'HH:MI:SS') begin_time
, TO_CHAR(end_time, 'HH:MI:SS') end_time
, instance_name
, elapsedpercall service_time
, callspersec throughput
FROM gv$instance i
, gv$active_services s
, gv$servicemetric m
WHERE s.inst_id = m.inst_id
 AND s.name_hash = m.service_name_hash
 AND i.inst_id = m.inst_id
 AND m.group_id = 10
 ORDER BY  service_name , i.inst_id , begin_time ;
```

Service Thresholds and Alerts

- For the services, you can set alerts for:
  - ELAPSED_TIME_PER_CALL Elapsed time for each user call for each service (micro s)
  - CPU_TIME_PER_CALL CPU time for each user call for each service (micro s)
- Thresholds must be set on each instance supporting the service

```
-- thresholds must be set on each instance supporting the service
begin
DBMS_SERVER_ALERT.SET_THRESHOLD(
     METRICS_ID => DBMS_SERVER_ALERT.ELAPSED_TIME_PER_CALL
   , WARNING_OPERATOR => dbms_server_alert.operator_ge
   , WARNING_VALUE => '500000' -- = 0.5 seconds
   , CRITICAL_OPERATOR => dbms_server_alert.operator_ge
   , CRITICAL_VALUE => '750000'   = 0.75 seconds
   , OBSERVATION_PERIOD => 30  -- in mins
   , CONSECUTIVE_OCCURRENCES => 5 -- tolerance occurance before alerts
   , INSTANCE_NAME => NULL -- must be NULL in this case
   , OBJECT_TYPE => dbms_server_alert.object_type_service
   , OBJECT_NAME => 'hrserv');
end;
/

-- Verify the threshold configuration
SELECT METRICS_NAME, INSTANCE_NAME, WARNING_VALUE, CRITICAL_VALUE,
OBSERVATION_PERIOD
from dba_thresholds
where  OBJECT_NAME => 'hrserv' ;

-- most recent 60 s
SELECT service_name, elapsedpercall, cpupercall
FROM V$SERVICEMETRIC;

-- last hour
SELECT service_name, elapsedpercall, cpupercall
FROM V$SERVICEMETRIC_HISTORY;
```

Service Performance Views

- There are more than 300 performance-related statistics that are tracked and visible in V$SYSSTAT. Of these, 28 statistics are tracked for services. To see the statistics measured for services, run the following query: SELECT DISTINCT stat_name FROM v$service_stats.
- For service metrics, see Monitoring Load Balancing Advisory

```
-- general stats per service
-- DB time (response time)= DB CPU + wait time
select V.SERVICE_NAME,
decode(n.CLASS,
'1','User','2','Redo','4','Enqueue','8','Cache','16','OS','32','RAC','64','SQL
','128','Debug', N.CLASS) STAT_CLASS
, V.STAT_NAME , VALUE
from   V$SERVICE_STATS V , V$STATNAME N
where  V.STAT_ID = N.STAT_ID
order by 1,2;


-- aggregated waits (by wait class)
SELECT  *
FROM    V$SERVICE_WAIT_CLASS
WHERE SERVICE_NAME NOT LIKE 'SYS$%'
ORDER BY SERVICE_NAME, TIME_WAITED DESC


-- aggregated waits (by wait event)
SELECT *
FROM V$SERVICE_EVENT
WHERE SERVICE_NAME NOT LIKE 'SYS$%'
AND EVENT NOT IN ('SQL*Net message from client')
ORDER BY SERVICE_NAME, TIME_WAITED DESC;


-- stats for a specific combination of service/module/action
-- When statistics collection for specific modules and actions is enabled
select *
from V$SERV_MOD_ACT_STATS
ORDER BY 1,2,3,4

-- information about enabled on-demand statistic aggregation
select *
from DBA_ENABLED_AGGREGATIONS;

-- information about enabled SQL traces
selct *
from DBA_ENABLED_TRACES;
```

Restricted Session and Services

- If the restricted instance is manually registered in the listener, the incoming connection requrests may be routed to the instance, refused by the instance and returns error.
- If the restricted instance is dynamically registered in the listener, the incoming connection requrests will not be routed to the instance, even for users with RESTRICTED SESSION privilege.

Configuring Recovery Manager and Archiving

Backup Possible Distributions in RAC

- **Network backup server**: dedicated backup server manages backups for the whole RAC.
- **One local drive**: one node has access to a local backup appliance.

```
/* Multiple Drives CFS Backup Scheme */
-- cfs = all nodes have their archived log in shared storage
-- one time setup
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'user1/passwd1@node1';
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'user2/passwd2@node2';
CONFIGURE CHANNEL 3 DEVICE TYPE sbt CONNECT 'user3/passwd3@node3';

-- backup is then distributed in all backup tapes in the cluster
BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

- **Multiple drives**: each node has access to a local backup appliance.

RMAN Restore Scenarios for Real Application Clusters

Cluster File System Restore Scheme

1. Make the Backup media and all the Archived Log files available to the available node.
2. Then Restore using RESOTRE DATABASE and RECOVER DATABASE commands.

Non-Cluster File System Restore Scheme

1. You must configure a network file system file so that the recovery node can read the archiving directories on the remaining nodes.
2. Then Restore using RESOTRE DATABASE and RECOVER DATABASE commands.

RMAN and Oracle Net in Real Application Clusters

- Connection server must be dedicated.
- Each net service name must specify only one instance.

Connecting to Specific Node

```
rman TARGET=SYS/ORACLE@RAC1 CATALOG=RMAN/RMAN@RMAN
```

Instance Recovery in Real Application Clusters

Single Node Failure in Real Application Clusters

- When one instance performs recovery for another instance, the surviving instance reads online redo logs generated by the failed instance and uses that information to ensure that committed transactions are recorded in the database.

Multiple-Node Failures in Real Application Clusters

- If all instances of an Oracle RAC database fail, then Oracle automatically recovers the instances the next time one instance opens the database.

Configuring the RMAN Snapshot Control File Location

- The snapshot control file is a temporary snapshot control file that RMAN creates to re-synchronize from a read-consistent version of the control file.

```
-- configured location of the snapshot control file
SHOW SNAPSHOT CONTROLFILE NAME;

-- to change it (globally in all nodes)
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '$ORACLE_HOME/dbs/scf/snap_prod.cf';
```

Configuring the RMAN Control File and SPFILE Autobackup Feature

```
# switch it on
CONFIGURE CONTROLFILE AUTOBACKUP ON;

# Auto control file and spfile backup must be configured in a shared location.
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT = '/u02/oradata/RAC/cf_%F';
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '+FRA';

# in the recovery destination
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F';
```

Configuring Channels for RMAN in Real Application Clusters

Configuring Channels to use Automatic Workload Balancing

```
CONFIGURE DEVICE TYPE [disk | sbt] PARALLELISM number of channels;

# configure the SQL Net service name with load balancing turned on
CONFIGURE CHANNEL DEVICE TYPE SBT CONNECT 'sys/pwd1@<service_name>'
```

Configuring Channels to Use a Specific Instance

```
# manual instance assignment
# To configure channels to use a specific instance, use the following syntax:
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'SYS/pss@rac1'
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'SYS/pss@rac2'

# Automatic instance assignment
```

```
# you can configure special service for rman jobs
#  service workload will be used
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
CONFIGURE CHANNEL DEVICE TYPE sbt CONNECT='sys/rac@rmanserv';
# also, parallel configuration will distribute the job on multiple instances
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;


# configuring channels manually
RUN
{
 ALLOCATE CHANNEL CH1 CONNECT 'user1/pwd1@node1';
 ALLOCATE CHANNEL CH2 CONNECT 'user2/pwd2@node2';
 ALLOCATE CHANNEL CH3 CONNECT 'user3/pwd3@node3';
 BACKUP DATABASE PLUS ARCHIVED LOG;
}
```

Node Affinity Awareness of Fast Connections

- When deciding which channel to use to back up a particular datafile, RMAN gives preference to the nodes with faster access to the datafiles that you want to back up.

Archived Redo Log File Conventions in RAC

```
show parameter LOG_ARCHIVE_FORMAT
alter system set LOG_ARCHIVE_FORMAT='log_%t_%s_%r.arc' scope=spfile sid='*';
```

Archive Redo Log Configuration Scenarios

Automatic Storage Management and CFS Archiving Scheme

- The preferred configuration for RAC is to use Automatic Storage Management (ASM) for a recovery area with a different disk group for your recovery set than for your datafiles.

```
show parameter archive_dest
show parameter recovery

# each node archives to a directory with the same name on all instances
sid1.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest"
sid2.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest"
sid3.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest"
```

Non-Cluster File System Local Archiving Scheme

- In the non-cluster file system local archiving scheme, each node archives to a uniquely named local directory. The directory in each node can be read only mounted from the other nodes.

```
# archived log stored locally and the other locations can be read by the node
#  (for recovery)
sid1.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest_1"
sid2.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest_2"
sid3.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest_3"
```

Changing the Archiving Mode in Real Application Clusters

```
select log_mode from v$database;

srvctl stop database -d rac
Back up the database
set LOG_ARCHIVE_DEST_n
show parameter LOG_ARCHIVE_DEST
-- no instance in OPEN mode
STARTUP MOUNT
ALTER DATABASE ARCHIVELOG;
SHUTDOWN IMMEDIATE
Back up the database
srvctl start database -d rac
```

Deleting Archived Redo Logs after a Successful Backup

```
# if automatic channels are configured
DELETE ARCHIVELOG ALL BACKED UP n TIMES TO DEVICE TYPE device_type;

# configuring the channels manually
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT 'SYS/oracle@node1';
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT 'SYS/oracle@node2';
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT 'SYS/oracle@node3';
DELETE ARCHIVELOG ALL BACKED UP n TIMES TO DEVICE TYPE device_type;
```

Monitoring the Archiver Processes

```
SELECT A.INST_ID, A.PROCESS, A.STATUS, A.LOG_SEQUENCE, A.STATE
FROM GV$ARCHIVE_PROCESSES A
WHERE STATUS<>'STOPPED'
ORDER BY A.INST_ID

# archived log files
SELECT INST_ID , THREAD#, NAME FROM GV$ARCHIVED_LOG;
```

Log_Archive_Dest_1 Set To Default Even When DB_Recovery_File_Dest Is Set (Bug 6373164)

You should take this bug into consideration on Oracle 11.1.0.6. If you just leave DEST_10 pointing to the flash recovery areas (as the default case), DEST_1 will also point to the default location: ORACLE_HOME\RDBMS

Here are the steps to reproduce the bug:

- When I disable LOG_ARCHIVE_DEST_1, the v$parameter and V$ARCHIVE_DEST are correctly updated. But after I restart the db, the LOG_ARCHIVE_DEST_1 is NULL in v$parameter but it points to C:\app\Administrator\product\11.1.0\db_1\RDBMS in V$ARCHIVE_DEST.
- DEST_10 is pointing to the recovery area but it's shown by V$ARCHIVE_DEST and not by v$parameter. I also noticed Oracle writes archive log files in the both destinations: DB_RECOVERY_FILE_DEST and the default one.

- **Work around**: Explicitly set LOG_ARCHIVE_DEST_1 to LOCATION=USE_DB_RECOVERY_FILE_DEST

```
SQL> select schedule, destination
  2  from V$ARCHIVE_DEST
  3  where dest_name='LOG_ARCHIVE_DEST_1';

SCHEDULE DESTINATION
-------- ------------------------------------------------
ACTIVE   C:\app\Administrator\product\11.1.0\db_1\RDBMS


SQL> select name, value
  2  from v$parameter
  3  where upper(name) like 'LOG_ARCHIVE_DEST_1';

NAME                           VALUE
------------------------------ ------------------
log_archive_dest_1


SQL> alter system set LOG_ARCHIVE_DEST_1='' scope=both;
System altered.



SQL> select schedule, destination
  2  from V$ARCHIVE_DEST
  3  where dest_name='LOG_ARCHIVE_DEST_1';

SCHEDULE DESTINATION
-------- ---------------------------------
INACTIVE


SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup
ORACLE instance started.

Total System Global Area  431038464 bytes
Fixed Size                  1333676 bytes
Variable Size             343934548 bytes
Database Buffers           79691776 bytes
Redo Buffers                6078464 bytes
Database mounted.
Database opened.


SQL> select name, value
  2  from v$parameter
  3  where upper(name) like 'LOG_ARCHIVE_DEST_1';

NAME                           VALUE
------------------------------ ------------------
```

```
log_archive_dest_1


SQL> select schedule, destination
  2  from V$ARCHIVE_DEST
  3  where dest_name='LOG_ARCHIVE_DEST_1';

SCHEDULE DESTINATION
-------- ---------------------------------------------
ACTIVE   C:\app\Administrator\product\11.1.0\db_1\RDBMS



SQL> select schedule, destination
  2  from V$ARCHIVE_DEST
  3  where dest_name='LOG_ARCHIVE_DEST_10';

SCHEDULE DESTINATION
-------- ----------------------------------
ACTIVE   USE_DB_RECOVERY_FILE_DEST

SQL> select name, value
  2  from v$parameter
  3  where upper(name) like 'LOG_ARCHIVE_DEST_10';

NAME                           VALUE
------------------------------ -------------------
log_archive_dest_10
```

Media Recovery in Real Application Clusters

- Same as standalone database.

Parallel Recovery in Real Application Clusters

- When parallelism is configured, Oracle automatically use the optimum degree of parallelism for recovery.

```
# Disabling Instance and Crash Recovery Parallelism
configure RECOVERY_PARALLELISM to 0

# Disabling Media Recovery Parallelism
RMAN> RECOVER DATABASE NOPARALLEL
SQL>ALTER DATABASE RECOVER NOPARALLEL
```

Using a Flash Recovery Area in RAC

- Set the parameter DB_RECOVERY_FILE_DEST to a shared location and must be the same value on all instances.

Managing Backup and Recovery

Administrative Options

Using Enterprise Manager Grid Control to Discover Nodes and Instances

To discover targets if a database is created after agents are installed or if a database is not automatically discovered at agent install time:

1. Log in to Enterprise Manager and click the **Targets** tab.
2. Click the **Database** tab to view all of the available targets. The column labeled **Types** shows the Oracle RAC databases using the entry "**Cluster Database**".
3. Select the **target name**, then clicking **Add**. The Add Database Target: Specify Host page appears, which enables you to add databases, Listeners, and Automatic Storage Management (ASM) as monitored targets.
4. Click the **flashlight icon** to display the available host names, select a host, then click **Continue**. The Add Database: Specify Source page appears.
5. Either request Enterprise Manager to discover only single-instance databases and Listeners, or to discover all cluster databases, single-instance databases, and Listeners on the cluster, then click **Continue**.
6. Enterprise Manager performs discovery to locate and display the cluster database and its associated instances. The Targets Discovered on Cluster page appears. If this procedure did not discover your reconfigured cluster database and all of its instances, you can use this page to manually configure your cluster databases and single-instance databases.

Additional Information About SQL*Plus in RAC

How SQL*Plus Commands Affect Instances

- Most SQL statements affect the current instance.
- ALTER SYSTEM SWITCH LOGFILE affects only the current instance.
  - To force a global log switch, use the ALTER SYSTEM ARCHIVE LOG CURRENT statement.
  - The INSTANCE option of ALTER SYSTEM ARCHIVE LOG enables you to archive each online redo log file for a specific instance.

Displaying Running Instances

```
SHOW INSTANCE

SELECT * FROM V$ACTIVE_INSTANCES;
```

Displaying Connect Identifier

```
use SQLPROMPT Command _CONNECT_IDENTIFIER
SET SQLPROMPT "_CONNECT_IDENTIFIER _USER > "
```

Quiescing RAC Databases

- The ALTER SYSTEM QUIESCE RESTRICTED and ALTER SYSTEM UNQUIESCE statements affect all instances in an Oracle RAC environment.

```
ALTER SYSTEM QUIESCE RESTRICTED;
```

- To set the database in the QUIESCE state, the Database Resource Manager must be activated in all the instances.

```
-- to make sure the resource manager is active (not null)
select value from v$parameter where upper(name)='RESOURCE_MANAGER_PLAN';
```

Quiesced State and Cold Backups

- You cannot use the quiesced state to take a cold backup.

Transparent Data Encryption and Wallets in RAC

- One wallet shared by all instances on shared storage.
- One copy of the wallet on each local storage: Local copies need to be synchronized each time master key is changed.

```
The wallet must reside in the directory specified by the setting for the
WALLET_LOCATION or ENCRYPTION_WALLET_LOCATION parameter in sqlnet.ora.
```

Administering System and Network Interfaces with oifcfg

Defining Network Interfaces with oifcfg

Syntax and Commands for the oifcfg Command-Line Tool

```
Oracle Interface Configuration

oifcfg -help

oifcfg iflist

oifcfg getif [ [-global | -node nodename] [-if if_name[/subnet]] [-type
if_type] ]

# store the interface hme0, with the subnet 139.185.141.0, as a global
# interface (to be used as an interconnect for all of the RAC instances in
your cluster)
oifcfg setif -global hme0/139.185.141.0:cluster_interconnect

# create the cms0 interface in rac1 and rac2
oifcfg setif -node rac1 cms0/139.185.142.1:cluster_interconnect
oifcfg setif -node rac2 cms0/139.185.142.2:cluster_interconnect

# deleting an interface
# CAUTION: when no option provided, all interfaces will be deleted:
oifcfg delif -global qfe0/204.152.65.16
# all global interfaces will be deleted
oifcfg delif -global
```

Changing Public or Interconnect IP Subnet Configuration

- A network interface can be stored as a global interface or as a node-specific interface. An interface is stored as a global interface when all the nodes of a RAC cluster have the same interface connected to the same subnet (recommended). It is stored as a node-specific interface only when there are some nodes in the cluster that have a different set of interfaces and subnets.

```
su -
$CRS_HOME/bin/oifcfg getif

oifcfg delif -global eth0
oifcfg setif –global eth0/139.2.166.0:public

oifcfg delif –global eth1
oifcfg setif –global eth1/192.168.1.0:cluster_interconnect

oifcfg getif
```

Changing VIP Addresses

- If only the IP address is changed, it is not necessary to make changes to the listener.ora, tnsnames.ora and initialization parameter files, provided they are using the virtual host names. If changing both the virtual host name and the VIP address for a node, it will be necessary to modify those files with the new virtual host name. For the listener.ora file,you can use netca to remove the old listener and create a new listener. In addition, changes will need to be made to the tnsnames.ora file of any clients connecting to the old virtual host name.

```
1. Confirm the current IP address for the VIP by running the ifconfig –a
command. On Windows, run the ipconfig /all command.
ifconfig –a

2. Stop First, the database instance, and then the ASM instance. When done,
stop nodeapps.
srvctl stop instance ..
srvctl stop asm ..
srvctl stop nodeapps

3. Verify that the VIP is no longer running by executing the ifconfig -a
command again:
ifconfig –a

If the interface still online, this is an indication that a resource which is
dependent on the VIP is still running. The crs_stat -t command can help to
show resources that are still online:
crs_stat -t

4. Make any changes necessary to all nodes' /etc/hosts files (on UNIX), or
\WINNT\System32\drivers\etc\hosts files on Windows, and make the necessary DNS
changes, to associate the new IP address with the old host name.

5. Modify nodeapps and provide the new virtual IP address:
su -
srvctl modify nodeapps -n rac1 -A 192.168.2.125/255.255.255.0/eth0

6. Start nodeapps again.
srvctl start nodeapps -n rac1
```

```
7. Repeat the same steps for all the nodes in the cluster. You can stay
connected from the first node.

8. verifies the connectivity between all of the nodes.
cluvfy comp nodecon -n all [-verbose]

9. Restart all of the instances and node applications
```

Adding Nodes and Instances on UNIX-Based Systems

Adding Nodes to a RAC Environment

In adding nodes to a cluster, there are three procedures that can be used:

- Cloning: copy images of Oracle Clusterware and RAC software onto the other nodes that have identical hardware and software.
- OEM: GUI of cloning.
- Interactive or silent procedures using scripts and DBCA.

Cloning Oracle Clusterware and RAC Software in Grid Environments

Prerequisites

- Make the required prerequisites changes on OS to prepare the node for the software cloning. The steps stated in the Preinstallation Tasks illustrates the steps required for Oracle 10g R2 for Oracle Linux Enterprise 4.5 for x86.
- Metalink document ID 169706.1 is quick reference of OS Installation and Configuration Requirements.
- The DB Console should be installed so that the required Perl files are installed in $ORACLE_HOME/clone/bin
- If the server has more than one Perl version installed, it may be necessary to specify the PERL5LIB environment variable so that the versions of the Perl modules match with the Perl version used.

Cloning Procedure Steps

Cloning Oracle10g Release 2 on Oracle Linux Enterprise 4.5 for x86 Using OUI:

1. To ensure the integrity of the copy, shutdown any databases, listeners, agents etc. that are running from the source home.
2. Clone the Oracle Clusterware home then Oracle Database home

```
cd /u01/crs
tar -cvf /tmp/sourcecrs.tar
cd /u01/app/oracle/product/10.2.0
tar -cvf /tmp/sourcedb.tar
```

3. Make sure the required users and groups are there on the new node

```
id oracle
id oinstall
```

4. Extract the tar file

```
cd /u01/crs
tar -xvf /tmp/sourcecrs.tar
cd /u01/app/oracle/product/10.2.0
tar -xvf /tmp/sourcedb.tar
```

5. Run the Oracle Universal Installer (OUI) in clone mode

```
/* Method 1 */
cd $ORACLE_HOME/clone/bin
perl clone.pl ORACLE_HOME="<target_home>" ORACLE_HOME_NAME="<unique_home_name>"
/* Method 2 */
cd $ORACLE_HOME/oui/bin
./runInstaller -clone -silent -ignorePreReq ORACLE_HOME="<target_home>"
ORACLE_HOME_NAME="<unique_home_name>"
```

If necessary, add "-invPtrLoc <path>/oraInst.loc" or "-ignoreSysPrereqs" to the command line.

6. Run the installation scripts
 `$ORACLE_HOME/root.sh`
7. Repeat steps 1 to 5 on Oracle home with RAC software
8. Run the Oracle Net Configuration Assistant (NETCA) on the new node to create a Listener.
9. Make if necessary, any modification in the tnsnames.ora file.
10. To start the dbconsole on the new server, refer to Metalink Note 467598.1 and Note 278100.1 for

Quick-Start Node and Instance Addition Procedures

Prerequisites

- Make the required prerequisites changes on OS to prepare the node for the software cloning.
  - The steps stated in Preinstallation Tasks illustrates the steps required for Oracle 10g R2 for Oracle Linux Enterprise 4.5 for x86. Be careful about the following changes in the steps:
    - The examples in this sections assumes the node to add has the ip addresses public:192.168.4.15, vip:192.168.4.17, priv:192.168.0.15 with hostname of rac3.
    - ORACLE_SID=rac3 in the .bash_profile file
    - Be careful NOT to partition the disks as they are already partitioned and used by the other node(s). However, raw disks must be binded.
    - Domain name resolution (/etc/hosts) must apply in all nodes
    - Remote installation procedure must be redone to all nodes.
- Metalink document ID 169706.1 is quick reference of OS Installation and Configuration Requirements for the supported Operating Systems.

Adding an Oracle Clusterware Home to a New Node

- Practically, I noticed the procedure does not work unless all the existing nodes in the RAC are alive.

```
# (1) run add node in rac1
# as oracle
export CRS_HOME=/u01/crs
cd $CRS_HOME/oui/bin
./addNode.sh
# Specify the New Nodes then Next
# Verify the new node names in the Summary page then Finish (copy size:80 MB)
# run the scripts as indicated by OUI. They must be:
# /u01/app/oracle/oraInventory/orainstRoot.sh  in rac3
# /u01/crs/install/rootaddnode.sh              in rac1
# in rac3: EDIT root.sh and fix the node list in its header BEFORE running it
# /u01/crs/root.sh                           in rac3
# exit the OUI

## if root.sh failed, you can cancel its effect by running:
# $CRS_HOME/install/rootdelete.sh

# (2) rac1 as oracle: run the Oracle Notification Service (RACGONS)
# obtain remote port number
cat $CRS_HOME/opmn/conf/ons.config
$CRS_HOME/bin/racgons add_config rac3:6200
#to verify
ping rac3-vip
crs_stat -t
```

```
# if you're using vmware, make backup of the environment now.
```

Adding an Oracle Home with RAC to a New Node

- OUI can be used interactively or in silent mode. This example shows the interactive mode steps.

```
# (1) copy Oracle home
# in rac1 as oracle
cd $ORACLE_HOME/oui/bin
./addNode.sh
# select node then Next
# verify info in the Summary page then Next
# about 1.5 GB will be copied to rac3
# run the required script

# (2) add Listener
# rac3 as oracle: run netca to add a listener in rac3 node (select rac3 ONLY)
$ORACLE_HOME/bin/netca

# (3) create db instance as descried in "Adding Database Instances to New
Nodes"

# (4) Perform Postinstallation Steps. Be aware to check the ASM instance name
before editing /etc/oractab file: ps -ef | grep asm

# (5) Review $ORACLE_HOME/network/admin/tnsnames.ora file and fix node
#      names errors, if any.

# (6) preferably perform General Useful Postinstallation Tasks.
```

Detailed Node and Instance Addition Procedure

Step 1: Connecting New Nodes to the Cluster

Making Physical Connections
Installing the Operating System
Creating Oracle Users
Verifying the Installation with the Cluster Verification Utility
Checking the Installation

Step 2: Extending Clusterware and Oracle Software to New Nodes

Adding Nodes at the Vendor Clusterware Layer
Adding Nodes at the Oracle Clusterware Layer

Step 3: Preparing Storage on New Nodes

Raw Device Storage Preparation for New Nodes

Step 4: Adding Nodes at the Oracle RAC Database Layer

Step 5: Adding Database Instances to New Nodes

Using Enterprise Manager to Add Database Instances to New Nodes

Using DBCA in Interactive Mode to Add Database Instances to New Nodes

1. Start dbca in rac1:

```
$ORACLE_HOME/bin/dbca &
```

2. Follow: Real Application Cluster-> Instance Management-> Add Instanc-> Select RAC database-> Next-> Enter the Instance Name, select node-> Enter service info->

```
$CRS_HOME/bin/crs_stat -t

srvctl start service -d rac -s "hrserv" -i rac3
srvctl start service -d rac -s "oeserv" -i rac3
```

Using DBCA in Silent Mode to Add Database Instances to New Nodes

Deleting Nodes and Instances on UNIX-Based Systems

Option 1: Quick-Start Node and Instance Deletion Procedures

- Oracle Database home is always deleted first, then the Clusterware home is deleted.
- Quick-Start node and instance deleteion procedure could be the convenient option when you want to delete one or two nodes.

Deleting an Oracle Home with RAC from an Existing Node

- The target here is to delete the Oracle database home. OUI can be used in interactive or silent mode to acheive this target. Interactive method is shown here.
- The example below assumes that you remove Oracle Database home from a node named rac2:

1. Remove rac2 from the available or preferred list of all the services:

```
# list the services
srvctl status service -d rac

# stop the services in rac2
srvctl stop service -d rac -s hrserv -i rac2
srvctl stop service -d rac -s oeserv -i rac2

# check rac2 is used by any service
srvctl config service -d rac

# my services available only in the remaining nodes
srvctl modify service -d rac -s hrserv -n -i rac1
srvctl modify service -d rac -s oeserv -n -i rac1

# confirm
srvctl config service -d rac
```

2. From rac1, remove DB instance in rac2 using DBCA (OEM can also be used). Follow the steps as in the sub-section "Using DBCA in Interactive Mode to Delete Database Instances from Existing Nodes". To confirm:

```
crs_stat -t
```

3. Delete ASM instance as described in the sub-section "ASM Instance Clean-Up Procedures for Node Deletion".
4. Use NETCA to remove the listener. To confirm:

```
crs_stat -t
```

5. Stop nodeapps from rac2 then remove them.

```
su -
cd /u01/crs/bin
./srvctl stop nodeapps -n rac2
./crs_stat -t
./srvctl remove nodeapps -n rac2
```

6. On rac2, to make the OUI remove the software home only from rac2:

```
su - oracle
echo $ORACLE_HOME
export ORACLE_HOME=/u01/app/oracle/product/10.2.0/db_1
```

```
cd $ORACLE_HOME/oui/bin
./runInstaller -updateNodeList ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES={rac2}"
–local
```

7. Run OUI from the home and deinstall this home. Make sure that you choose the home to be removed and not just the products under that home.

```
./runInstaller &
```

8. On rac1, run the following command where "CLUSTER_NODES={*remaining_list*}":

```
echo $ORACLE_HOME
export ORACLE_HOME=/u01/app/oracle/product/10.2.0/db_1
cd $ORACLE_HOME/oui/bin
./runInstaller -updateNodeList ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES={rac1}"
```

Deleting an Oracle Clusterware Home from an Existing Node

- The target here is to delete the Oracle Clusterware home from a node. OUI can be used in interactive or silent mode to acheive this target. Interactive method is shown here.
- The example below assumes that you remove Oracle Clusterware home from a node named as rac2:

1. Makre sure Oracle Database home was removed before going on the clusterware home deletion procedure.

2. If you ran the Oracle Interface Configuration Tool (OIFCFG) with the -global flag during the installation, then skip this step. Otherwise, from a node that is going to remain in your cluster, from the CRS_home/bin directory, run the following command:

```
./oifcfg delif –node rac2
```

3. Obtain the remote port number, which you will use in the next step, using the following command from the CRS_HOME/opmn/conf directory:

```
echo $CRS_HOME
export CRS_HOME=/u01/crs
cd $CRS_HOME/opmn/conf
cat ons.config
```

4. In rac1: run the Oracle Notification Service (RACGONS)

```
cd $CRS_HOME/bin
./racgons remove_config rac2:remote_port
```

5. On rac2 as root:

```
su -
export CRS_HOME=/u01/crs
$CRS_HOME/install/rootdelete.sh
```

6. On rac1 as root: run `./rootdeletenode.sh node2,node2-number`

```
su -
export CRS_HOME=/u01/crs
# if you want to identify the node number
$CRS_HOME/bin/olsnodes -n

$CRS_HOME/install/rootdeletenode.sh rac2,2
$CRS_HOME/bin/olsnodes -n
```

7. On rac2:

```
su - oracle
export CRS_HOME=/u01/crs
cd $CRS_HOME/oui/bin
./runInstaller -updateNodeList ORACLE_HOME=$CRS_HOME "CLUSTER_NODES={rac2}"
CRS=TRUE -local
```

8. On rac2, de-install the Oracle Clusterware using the OUI:

```
cd $CRS_HOME/oui/bin
./runInstaller &
```

9. On rac1, where "CLUSTER_NODES={remaining_nodes_list}":

```
su - oracle
cd $CRS_HOME/oui/bin
./runInstaller -updateNodeList ORACLE_HOME=$CRS_HOME "CLUSTER_NODES={rac1}"
CRS=TRUE

crs_stat -t
olsnodes -n
```

Option 2: Detailed Node and Instance Deletion Procedure

- Detailed node and instance deleteion procedure could be more convenient when you want to delete high number of nodes.

Step 1: Deleting DB Instances from Real Application Clusters Databases

Using Enterprise Manager to Delete Database Instances from Existing Nodes

- Follow the links: Cluster Database Home page-> Maintenance-> Deployments section-> Delete Instance-> Deletion Wizard starts

Using DBCA in Interactive Mode to Delete Database Instances from Existing Nodes

1. On rac1, open DBCA from Oracle home.
2. On the DBCA Welcome page select Oracle Real Application Clusters Database, click Next
3. On the DBCA Operations page, select Instance Management, click Next
4. On the Instance Management page, Select Delete Instance, click Next
5. On the List of Cluster Databases page, select the Oracle RAC database. Enter a SYSDBA user name and password. Click Next
6. On the List of Cluster Database Instances page, select an instance to delete and click Finish.
7. If you have services assigned to this instance, then the DBCA Services Management page appears. Use this feature to reassign services from this instance to other instances in the cluster database.
8. On the Summary page and click OK.
9. DBCA removes the instance and the instance's Oracle Net configuration.
10. After completion, exit DBCA

Using DBCA in Silent Mode to Delete Instance from Existing Nodes

1. General Syntax:

dbca -silent -deleteInstance [-nodeList *node*] -gdbName gdbname –instanceName *instname* -sysDBAUserName *sysdba* -sysDBAPassword *password*

```
-- if  running form rac2:
dbca -silent -deleteInstance -gdbName rac -instanceName rac2 -sysDBAUserName
sys -sysDBAPassword syspassword

-- if running from rac1:
dbca -silent -deleteInstance -nodeList rac2 -gdbName rac -instanceName rac2 -
sysDBAUserName sys -sysDBAPassword syspassword
```

Step 2: Deleting Nodes from Real Application Clusters Databases

- The example assumes rac2 is the node to delete and that Oracle home is local (not shared).

1. If there are instances on the node that you want to delete, then perform one of the procedures in the previous section “Setp1”.
2. Perform ASM clean up procedure as described in the sub-section "ASM Instance Clean-Up Procedures for Node Deletion".
3. Use NETCA to remove the listener from rac2.
4. On rac2:

```
cd $ORACLE_HOME/oui/bin
runInstaller -updateNodeList ORACLE_HOME=$ORACLE_HOME CLUSTER_NODES="" –local
```

5. Run OUI from the home and deinstall this home. Make sure that you choose the home to be removed and not just the products under that home.

```
./runInstaller &
```

6. On rac1, where "CLUSTER_NODES={node_list}":

```
runInstaller -updateNodeList ORACLE_HOME=Oracle_home "CLUSTER_NODES={rac1}"
```

7. Remove node-specific interface configurations:

```
# to obtain port number
echo $CRS_HOME
export CRS_HOME=/u01/crs
cd $CRS_HOME/opmn/conf
cat ons.config

racgons remove_config rac2:<remote_port>
oifcfg delif -node rac2
```

8. On rac2, disable the Oracle Clusterware applications that are on the node:

```
echo $CRS_HOME
export CRS_HOME=/u01/crs
cd $CRS_HOME/install
# If the ocr.loc file is on a local file system
rootdelete.sh remote nosharedvar
# If the ocr.loc file is on a shared file system
rootdelete.sh remote sharedvar
```

9. On any remaining node in the cluster (rac1), to delete the nodes from the Oracle cluster and to update the Oracle Cluster Registry (OCR):

```
su -
echo $CRS_HOME
export CRS_HOME=/u01/crs
# to display node numbers
$CRS_HOME/bin/olsnodes -n.

# to delete the node
# general syntax: rootdeletenode.sh node1,node1-number,node2,node2-number,...
$CRS_HOME/install/rootdeletenode.sh rac2,2
```

10. On rac2:

```
$CRS_HOME/oui/bin/runInstaller -updateNodeList ORACLE_HOME=$CRS_HOME
CLUSTER_NODES="" –local CRS=true
```

11. Run OUI from the home and deinstall this home. Make sure that you choose the home to be removed and not just the products under that home.

```
$CRS_HOME/oui/bin/runInstaller &
```

12. On rac1, where "CLUSTER_NODES={ramining_nodelist}":

```
runInstaller -updateNodeList ORACLE_HOME=CRS_home "CLUSTER_NODES=rac1"
```

13. Verify node removal:

```
cluvfy comp crs -n all [-verbose]
```

ASM Instance Clean-Up Procedures for Node Deletion

The target here is to remove ASM instance from a node.

1. Stop, if any, all of the databases that use the ASM instance. If you follow this procedure as part of a node deletion, the Db instance must have been removed by this step.
2. On rac2:

```
srvctl stop asm -n rac2

srvctl remove asm -n rac2

srvctl config asm -n rac2

srvctl config asm -n rac1

crs_stat -t
```

3. On rac2:

```
rm -r $ORACLE_BASE/admin/+ASM
rm -f $ORACLE_HOME/dbs/*ASM*
```

Adding and Deleting Nodes and Instances on Windows-Based Systems

Cloning Oracle Clusterware and RAC Software in Grid Environments

Quick-Start Node and Database Instance Addition and Deletion Procedures

Adding an Oracle Clusterware Home to a New Node

Adding an Oracle Home with RAC to a New Node

Deleting an Oracle Home with RAC from an Existing Node

Deleting an Oracle Clusterware Home from an Existing Node

Detailed Node and Database Instance Addition and Deletion Procedures

Overview of Node Addition Procedures

Step 1: Connecting New Nodes to the Cluster

Making Physical Connections

Installing the Operating System

Verifying the Installation with the Cluster Verification Utility

Checking the Installation

Step 2: Extending Oracle Software to New Nodes at the Oracle Clusterware

Step 3: Preparing Storage on New Nodes

Raw Device Storage Preparation for New Nodes

Step 4: Adding Nodes at the Oracle RAC Database Layer

Step 5: Adding Database Instances to New Nodes

Using Enterprise Manager to Add Database Instances to New Nodes

Using DBCA in Interactive Mode to Add Database Instances to New Nodes

Using DBCA in Silent Mode to Add Database Instances to New Nodes

Connecting to iSQL*Plus after Adding a Node

Adding Nodes that Already Have Clusterware and Oracle Software to a Cluster

Overview of Node Deletion Procedures

Step 1: Deleting Instances from Real Application Clusters Databases

Using Enterprise Manager to Delete Database Instances from Existing Nodes

Using DBCA in Interactive Mode to Delete Database Instances from Existing Nodes

Using DBCA in Silent Mode to Delete Instance from Existing Nodes

Step 2: Deleting Nodes from Real Application Clusters Databases

Step 3: ASM Instance Clean-Up Procedures for Node Deletion

Monitoring Performance

RAC Common Tuning Tips

- Application tuning is often the most beneficial
- Resizing and tuning the buffer cache
- Increasing sequence caches to a high value: to avoid index leaf contention caused by high or batch inserts.
- Reducing long full-table scans in OLTP systems: to reduce GCS requests.
- Using Automatic Segment Space Management
- Using partitioning to reduce interinstance traffic
- Avoiding unnecessary parsing
- Minimizing locking usage
- Removing unselective indexes
- Configuring interconnect properly

Instance Recovery and RAC

- _FAST_START_INSTANCE_RECOVERY_TARGET initialization parameters controls the time in seconds from the start of instance recovery to the time when GCD is open for lock requests.
- If FAST_START_MTTR_TARGET is also configured, the more aggressive will be used.
- Practically, using either of them is useless because one of the surviving instances is doing the recovery. You can however consider the following:
  - Increase PARALLEL_EXECUTION_MESSAGE_SIZE (message size for parallelism and its buffer taken from shared pool) from its default of 2,148 bytes to 4 KB or 8 KB.
  - Set PARALLEL_MIN_SERVERS to CPU_COUNT-1. This will prespawn recovery slaves at startup time.
  - Using asynchronous I/O is one of the most crucial factors in recovery time. The first-pass log read uses asynchronous I/O.
  - Instance recovery uses 50 percent of the default buffer cache for recovery buffers. If this is not enough, you will see it in the alert.log. In that case, increase the size of your default buffer cache.

Global Cache Wait Events

- gc current/cr request
- gc [current/cr] [2/3]-way
- gc [current/cr] block busy
- gc [current/cr] grant 2-way: grant occurs before I/O
- gc current grant busy
- gc [current/cr] [block/grant] congested: congested means it is internally > 1ms
- gc [current/cr] [failure/retry]
- gc buffer busy

Monitoring Performance in Enterprise Manager

Using the Cluster Database Performance Page

Chart: Cluster Host Load Average

Desc.: load on CPU in all cluster nodes

Vertical: No of CPUs in all the cluster

Notice : If the load average is higher than the average of the total number of CPUs across all of the hosts in the cluster, then too many processes are waiting for CPU resources.

Chart: Global Cache Block Access Latency

Desc.: end-to-end elapsed time or latency for a block request

Vertical: latency in ms

Notice : - interconnect delay - unoptimized SQL plans to acheive local cache hit ration. - to resolve: drill down to the Cluster Cache Coherency page

Chart: Global Cache Block Transfer Rate

Desc.: number of data blocks received by all instances

Notice: drid down till segment type

Chart: Average Active Sessions

Desc.: average number of active sessions per wait class.

Notice: Consider tuning the database, if the Average Active Sessions chart displays a large number of sessions waiting, indicating internal contention, and throughput the Database Throughput charts is low.

Chart : Database Throughput Charts

Desc. : summarize any contentions that appear in the Average Active Sessions chart

Notes : drill down to instance level then top consumers

Chart : Top Segments Page

Desc. : to identify hot tables or indexes in a database

Notes : tracks the number of CR and current blocks received by an object.

Chart : Database Locks Page

Desc. : to determine whether multiple instances are holding locks for the same object.

Using the Cluster Database Instance Performance Page

To access it, Performance tab-> instance name in the bottom.

Instance-level statistics including generating ADDM and ASH reports.

AWR: gather statistics about each individual instance in the RAC and not the entire database.

ADDM Report: create a new AWR snapshot and run ADDM on this and the previous snapshot.

```
exec dbms_advisor.set_default_task_parameter('ADDM','DB_ACTIVITY_MIN',30)
exec dbms_workload_repository.modify_snapshot_settings(interval=>600)

exec dbms_workload_repository.create_snapshot
```

ASH Report: create a performance data report of the database based on session-sampling data over a specified period of time. This report is very useful for diagnosing small (five- to ten-minute) performance spikes that might be averaged out and consequently hidden or minimized by other issues in the 1-hour AWR report.

Using the Cluster Performance Page

With this information, you can determine whether resources need to be added, suspended, or redistributed.

Using the Cluster Interconnects Page

Home page-> Interconnect Findings

Home page-> Performance-> Cluster Cache Coherency-> Interconnects

This page helps determine the load added by individual instances and databases on the interconnect. Sometimes you can immediately identify interconnect delays that are due to applications that are outside Oracle Database.

Making Applications Highly Available Using Oracle Clusterware

Note: You can install the Oracle Clusterware high availability Application Programming Interface (API) from the Oracle Database 10g release 10.2 client installation media.

Note: Your Listener.ora file should contain one entry for the virtual internet protocol (VIP) address, using the VIP's name, and another entry for the physical host, by IP address not name.

Making an Application Highly Available Examples

Example1: Making an Application Highly Available

- If the application is accessed by network, define a VIP per application.
- VIP is failed over to another node together with the application(s).
- From there, clients can still connect through the VIP.

1. Create an application profile by editing an ASCII file or by running the crs_profile command.

```
# file name format resource_name.cap
save file in
$CRS_HOME\crs\profile    for root user
$CRS_HOME\crs\public     for non-root user

# Required Ones
# resource-specific script (start, stop or check will be passed by crs)
ACTION_SCRIPT
# application name
NAME
# resource type (must be APPLICATION)
TYPE
# ordered list of cluster nodes (use names as listed by olsnodes)
HOSTING_MEMBERS

# Optional Ones
# description of the resource
DESCRIPTION='myapplication'
# list of required resources (must be registered)
REQUIRED_RESOURCES
# placement policy: (balanced), favored, or restricted
PLACEMENT
# when 1 (0), crs re-evaluates the placement of a resource
#  during addition or restart of a cluster node
ACTIVE_PLACEMENT=0
# when 1 (0), it atuo-restarts on system reboot
AUTO_START=1
# check intervals in seconds (60)
```

CHECK_INTERVAL
failover interval (0)
FAILOVER_DELAY
interval (s) during which crs applies the failure threshold
FAILURE_INTERVAL
(max 20) number of failures within FAILURE_INTERVAL after which
the resource is marked offline
FAILURE_THRESHOLD
space-delimited liste of resource used during placement decisions
OPTIONAL_RESOURCES
restart attempts before relocate (1)
RESTART_ATTEMPTS
maintained by crs
RESTART_COUNT
timeout in seconds the script needs before returning an error (60)
SCRIPT_TIMEOUT
START_TIMEOUT
STOP_TIMEOUT
application up-time to be considered stable by crs
UPTIME_THRESHOLD

```
# example 1
oracle$crs_profile -create postman -t application -B
/opt/email/bin/crs_postman \
-d "Email Application" -r network1 -l application2 \
-a postman.scr -o ci=5,ft=2,fi=12,ra=2

# in $CRS_HOME/crs/public
# will generate file  postman.cap containing:
NAME=postman
TYPE=application
ACTION_SCRIPT=/oracle/crs/script/postman.scr
ACTIVE_PLACEMENT=0
AUTO_START=0
CHECK_INTERVAL=5
DESCRIPTION=email app
FAILOVER_DELAY=0
FAILURE_INTERVAL=12
FAILURE_THRESHOLD=2
HOSTING_MEMBERS=
OPTIONAL_RESOURCES=application2
PLACEMENT=balanced
REQUIRED_RESOURCES=network1
RESTART_ATTEMPTS=2
SCRIPT_TIMEOUT=60
...
# and script file postman.scr

# Create Application VIP
# create network1 VIP address application
#  where eth0 is the public network nic
#   138.3.83.78 is the vip address
crs_profile –create network1 -t application \
```

```
-a $CRS_HOME/bin/usrvip \
-o oi=eth0,ov=138.3.83.78,on=255.255.240.0

# as oracle, register network1
oracle$crs_register network1

# change the owner of the resource
crs_setperm network1 –o root

# enable oracle user to run the script
crs_setperm network1 –u user:oracle:r-x

# as oracle starts the vip
crs_start network1
```

2. Register the application profile using the crs_register command.

```
crs_register postman
# If you modify postman profile, then update the OCR:
crs_register -u postman
```

3. Run the crs_start command to initiate the application profile and then the Oracle Clusterware runs the action program command that you have included in the profile to start your application.

```
crs_start postman

# to start an application resource, even if one of the required resources
#  is offline:
crs_start -f postman
```

4. The Oracle Clusterware periodically runs the action program command to check an application's status.

5. In the event of a check or node failure, the Oracle Clusterware recovers the applications either by restarting it on the current node or by relocating the application to another node. To manually relocate the application and its resources to another node:

```
# the application and its required resources must be offline
crs_relocate postman -c rac2
```

6. If you run the crs_stop command to stop the application, then the Oracle Clusterware runs the action program command to stop it.

```
crs_stop postman

# to stop an application that is required by an online resource
crs_stop -f postman
```

Example2: Making an Application Highly Available

```
/* Making xclock program highly available */

#
# 1) create the action script file
#  create following script in both nodes (in $CRS_HOME/crs/script)
#  make the file executable
#  you can then test it by passing start stop and check
```

```
su -
export CRS_HOME=/u01/crs
vi $CRS_HOME/crs/script/crsclock_action.scr
chmod 777 $CRS_HOME/crs/script/crsclock_action.scr

#!/bin/bash
# start/stop/check script for xclock example
#  the script assumes xclock is there
#  and DISPLAY variable is set
APP=/usr/X11R6/bin/xclock
BIN_NAME=xclock
LOG_DIR=/tmp

export DISPLAY=:0.0

echo `date +"%M:%S"` $0 $* $$>>/tmp/mylog.log

PID1=`ps -ef | grep $BIN_NAME | grep -v grep | grep -v xclock_app | awk '{
print $2 }'`

case $1 in
 'start')
  if [ "$PID1" != "" ]
  then
   status_p1="running"
  else
    if [ -x $APP ]
    then
     #umask 002
     ${APP} &
     PID1=`ps -ef | grep $BIN_NAME | grep -v grep | grep -v xclock_app | awk
'{ print $2 }'`
     echo `date +"%M:%S"` $* $PID1 $USER>>/tmp/mylog.log
     status_p1="started"
    else
     echo `basename $0`": $APP: Executable not found"
    fi
  fi
  echo "$APP: $status_p1"
  ;;
 'stop')
  if [ "${PID1}" != "" ]
  then
   kill -9 ${PID1} && echo "$APP killed"
  else
   echo "$BIN_NAME: no running Process!"
  fi
  ;;
 'check')
  if [ "$PID1" != "" ]
  then
   echo "running"
   exit 0
  else
   echo "not running"
   echo `date +"%M:%S"` $0 $* "ERR">>/tmp/mylog.log
   exit 1
```

```
  fi
  ;;
 *)
  echo "Usage: "`basename $0`" {start|stop|check}"
 ;;
 esac


#
# 2) create application resource profile named myClock
# if you copy paste, fix the hyphen issue
su -
$CRS_HOME/bin/crs_profile -create myClock -t application -a
crsclock_action.scr -p favored -h "rac1-vip rac2-vip" -o ci=5,ra=2

# check the generated file
cat $CRS_HOME/crs/profile/myClock.cap

#
# 3) Register the resource (in ONE node only)
#    must be done as root and then permission is granted to oracle
su -
$CRS_HOME/bin/crs_register myClock
$CRS_HOME/bin/crs_setperm myClock -u user:oracle:r-x
$CRS_HOME/bin/crs_stat myClock
$CRS_HOME/bin/crs_stat -t

#
# 4) Start the resource
#    as oracle
su - oracle
$CRS_HOME/bin/crs_start myClock
$CRS_HOME/bin/crs_stat -t
```

Managing Automatic Oracle Clusterware Resource Operations for Action Scripts

```
/* Preventing Automatic Database Instance Restarts */
In the application profile, set AUTO_START to (lowercase only):
always, restore, never

/* Automatically Manage Restart Attempts Counter for Resources */
This is controlled by the attributes RESTART_ATTEMPTS and RESTART_COUNT.
```

Displaying Clusterware Application and Application Resource Status Information

```
# application status
crs_stat postman

# all resources status
crs_stat -t
# specific resource status
crs_stat -t ora.rac1.ASM1.asm

# stats about the resources ( R=Restart, F=Fail)
crs_stat -v

# application profile
crs_stat -p
```

```
crs_stat -p ora.rac1.ASM1.asm
```

Unregistering Applications and Application Resources

```
su -
$CRS_HOME/bin/crs_unregister postman
$CRS_HOME/bin/crs_stat -t
```

RAC Troubleshooting

Diagnosing the Oracle Clusterware High Availability Components

Debugging Recommnedation

- Always make sure that your nodes have exactly the same system time. Use NTP.

Clusterware Log Files and the Unified Log Directory Structure

```
# Cluster Ready Services Daemon (crsd)
#  It is archived every 10 MB (crsd.l01, crsd.l02, …)
$CRS_HOME/log/hostname/crsd

# Oracle Cluster Registry (OCR)
$CRS_HOME/log/hostname/client

# Cluster Synchronization Services (CSS)
#  is archived every 20 MB (cssd.l01, cssd.l02, …)
$CRS_HOME/log/hostname/cssd

# Event Manager (EVM)
$CRS_HOME/log/hostname/evmd

# RACG Log Files
$CRS_HOME/log/hostname/racg
$ORACLE_HOME/log/hostname/racg

# SRVM (srvctl) and OCR (ocrdump, ocrconfig, ocrcheck) logs
$ORA_CRS_HOME/log/<hostname>/client

# crs alerts
$ORA_CRS_HOME/log/<hostname>/alert<nodename>.log
```

Dynamic Debugging

```
/* to Enable Debugging */
# as root
# for the Oracle Clusterware
crsctl debug log crs "CRSRTI:1,CRSCOMM:2"

# for EVM
crsctl debug log evm "EVMCOMM:1"

# for resources (1 is the debugging level)
crsctl debug log res "resname:1"
# example:
crsctl debug log res "ora.rac1.vip:1"
```

Component Level Debugging

```
# to enable debugging for all of the modules
# where level 1 (least) to 5 (max)
set ORA_CRSDEBUG_ALL <level>

# to enable tracing for a specific sub-module
```

```
set ORA_CRSDEBUG_modulename
# to list the sub-modules
# where module is crs, evm, or css
crsctl lsmodules <module>
```

Oracle Clusterware Shutdown and Startup

```
su -
crsctl stop crs
crsctl start crs
```

Enabling and Disabling Oracle Clusterware Daemons

```
su -
crsctl enable crs
crsctl disable crs
```

Diagnostics Collection Script

- Generates the following files in the local directory:

  basData_<hostname>.tar.gz
  crsData _<hostname>. tar.gz
  ocrData _<hostname>. tar.gz
  oraData _<hostname>. tar.gz

```
# when asked by Oracle support
su -
export ORACLE_HOME=/u01/app/oracle/product/10.2.0/db_1
export ORA_CRS_HOME=/u01/crs1020
export ORACLE_BASE= =/u01/app/oracle
cd $ORA_CRS_HOME/bin
$CRS_HOME/bin/diagcollection.pl -collect
```

The Oracle Clusterware Alerts

```
# can be found in:
$CRS_Home/log/hostname/alerthostname.log
```

Resource Debugging

```
# method 1: where 1 in the following example is the debugging level
crsctl debug log res "ora.node1.vip:1"

# method 2:
export USER_ORA_DEBUG=1
# .. then issue crsctl start, stop, or check
```

Checking the Health of the Clusterware

```
crsctl check crs
```

Troubleshooting the Oracle Cluster Registry

```

```

Troubleshooting Hostname Changes and CSS

If you change the host name for ASM, then the Oracle CSS daemon will not start. In order to counter this problem, please use the following steps:

- Login as the root user
- Run localconfig delete to deconfigure CSS. This will remove any configuration related files on the system that referenced the old host name.
- Run localconfig add to reconfigure CSS using the new host name.

Enabling Additional Tracing for Real Application Clusters High Availability

Diagnosing Oracle Real Application Clusters Components

Where to Find Files for Analyzing Errors

Using Instance-Specific Alert Files in Real Application Clusters

Enabling Tracing for Java-Based Tools and Utilities in Real Application Clusters

Resolving Pending Shutdown Issues

Using the Cluster Verification Utility

Cluster Verify Locations

- Download it from OTN
- Oracle software DVD:

  ```
  ./cluvfy/runcluvfy.sh
  ```
- Oracle Clusterware home:

  ```
  $ORA_CRS_HOME/bin/cluvfy
  ```
- Oracle Home:

  ```
  $ORACLE_HOME/bin/cluvfy
  ```

Cluster Verify Stages

- You can list verifiable stages with the `cluvfy stage -list`
- Valid stage options and stage names are (ordered):

  ```
  -post hwos: Postcheck for hardware and operating system
  -pre cfs: Precheck for OCFS setup
  -post cfs: Postcheck for OCFS setup
  -pre crsinst: Precheck for CRS installation
  -post crsinst: Postcheck for CRS installation
  -pre dbinst: Precheck for database installation
  -pre dbcfg: Precheck for database configuration
  ```

Cluster Verify Components

- You can list verifiable CVU components with the cluvfy comp -list
- Verifiable CVU components:

```
nodereach: Checks reachability between nodes
nodecon: Checks node connectivity
cfs: Checks Oracle Cluster File System integrity
ssa: Checks shared storage accessibility
space: Checks space availability
sys: Checks minimum system requirements
clu: Checks cluster integrity
clumgr: Checks cluster manager integrity
ocr: Checks OCR integrity
crs: Checks CRS integrity
nodeapp: Checks node applications existence
admprv: Checks administrative privileges
peer: Compares properties with peers
```

CVU Component Verification Examples

```
#verify the minimal system requirements on the nodes before installing
Clusterware
cluvfy comp sys -n node1,node2 -p crs -verbose


#check the system requirements before installing RAC:
cluvfy comp sys -n node1,node2 -p database -verbose


#verify whether storage is shared among the nodes in your cluster database or
to identify all of the storage that is available on the system and can be
shared across the cluster nodes:
cluvfy comp ssa -n all -s /dev/sda1


#check there is 5 GB free in all nodes:
cluvfy comp space -n all -l /home/product -z 5G


#can node1 reach node2:
cluvfy comp nodereach -n node2 -srcnode node1


#checks whether node1 and node2 can communicate through the eth0 network
interface (without i, all interfaces are checked):
cluvfy comp nodecon -n node1,node2 –i eth0 -verbose


#verify user equivalence for all the nodes:
cluvfy comp admprv -n all -o user_equiv -verbose


#verify existence of node applications, namely VIP, ONS, and GSD, on all
the nodes:
cluvfy comp nodeapp -n all -verbose
```

```
#compares all the nodes and determines whether any differences exist between
the values of preselected properties
cluvfy comp peer -n all –verbose | more
```

Understanding CVU Commands, Help, Output, and Nodelist Shortcuts

Performing Various CVU Tests

Known Issues for the Cluster Verification Utility

Part 8 Oracle RAC One Node

Installing Oracle 11g R2 RAC One Node on Linux 5

Overview

This installation implements Oracle RAC One Node. Oracle RAC One Node is a single instance of Oracle RAC running on a node in a cluster. RAC One Node offers a form of virtualization where multiple databases can be consolidated on the same physical server, sharing a common pool of storage and server resources and thus reducing the physical IT footprints.

Note: The metalink document RAC Starter Kit and Best Practices (Linux) [ID 811306.1] is a good source reference for this task.

Note: The procedure applies on 11.2.0.1. Version 11.2.0.2 provides simplified procedure.

Installation Environment

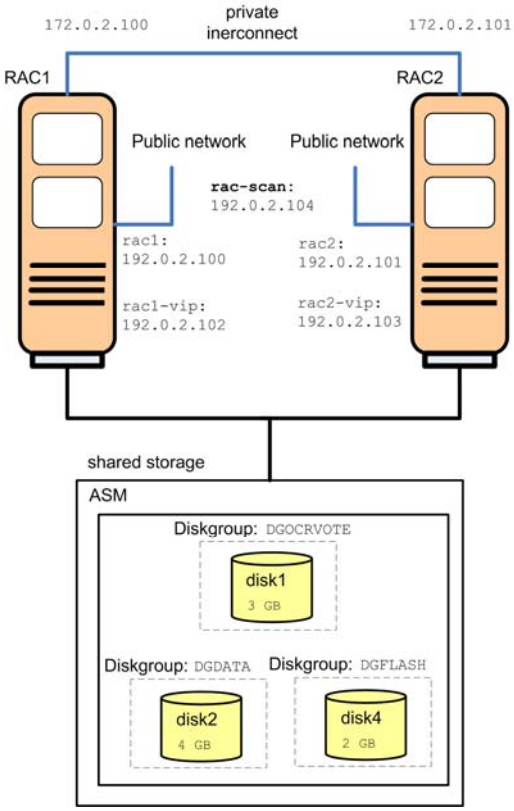
- Emulation software: VMWare Workstation 7
- RAC Node: 2 node with 2.5 GB RAM and 2 ethernet cards.
- OS: Red Hat Linux Enterprise 5.2 for x86 32-bit

Required Software

- Oracle Database 11g Release 2 for Linux x86 32-bit
- Oracle Database 11g Release 2 Grid Infrastructure (11.2.0.1.0) for Linux x86 32-bit
- RAC One Node patch# 9004119 : RACONENODE_p9004119_112010_LINUX.zip

Used Hardware

- In the VMWare: create one virtual machine (rac1) with the following specs:
  - 2.5 GB RAM
  - Two ethernet cards: both can be configured as bridged or host-only in VMware.
  - One local hardisk with 24 GB on SCSI 0:0.
  - CPU Count: 2
  - Create a folder in the same directory structure level as the parent folder containing the created virtual machine. Give it a meaningful name like 'shared_disks'. Create in that folder the following disks:
    - Disk1: of 3 GB. Allocate its disk space. It will be used for OCR and Voting disk. Set it on controller SCSI 1:1
    - Disk2: of 4 GB. Allocate its disk space. It will be used for +Data. Set it on controller SCSI 1:2
    - Disk3: of 2 GB. Allocate its disk space. It will be used for +Flash. Set it on controller SCSI 1:3



Installation Plan

1. Preinstallation tasks
    - Hardware requirements
    - Software requirements
    - Environment configuration
2. Oracle Grid Infrastructure installation
3. Oracle Grid Infrastructure Patching
4. Checking Oracle Grid Infrastructure Status
5. Oracle Database 11g R2 Software Installation
6. Oracle Database 11g R2 Software Patching
7. Install EM Agent in cluster nodes (if required)
8. ASM Diskgroups Creation
9. RAC Database Creation
10. Initialize the Database to RAC One Node
11. Complete postinstallation tasks
12. Useful postinstallation tasks

Note: The installation is explained without GNS and IPMI

Note: For this installation we will be using ASM for Clusterware and Database storage

1. Preinstallation tasks

- Install Oracle Enterprise Linux in the first local hardisk. Install nothing in the remaining disks.

  Note: for a production system, consider becoming an Oracle Unbreakable Linux customer and register your server on the Unbreakable Linux Network.

  - Configure the swap area in the local hardisk to have 3 GB disk space.
  - Give the first ethernet card IP 192.0.2.100 and the second 172.0.2.100 and the hostname `rac1.mydomain.com`. Define a gateway. If it does not exist, make it same as the host IP address.
  - Insall the following packages:
    - Desktop Environments
      - GNOME Desktop Environment
    - Applications
      - Graphical Internet (optional)
      - Editors (optional)
    - Development
      - Development Libraries
      - Development Tools
    - Servers
      - Do not select anything in this group.
    - Base System
      - Administration Tools
      - System Tools
        - Add the package 'sysstat' by clicking on the Details link and selecting "sysstat - The sar an iostat system monitoring commands." from the Optional Packages list.
    - X Window System
- Complete the installation.
- After the Installation compelets, RHEL 5.2 and below will hang on booting when it reaches to "starting udev" line. To solve this problem, shutdown the Vmware machine and change the CPU count and Core Count to only one. Implement the changes below, then shutdown the machine, set CPU count back to 2 and startup the machine.

  put the kernel command line parameters at the end of the "kernel" line:

  `vi /boot/grub/grub.conf`

  add `divider=10 clocksource=acpi_pm`

  For example: `kernel /vmlinuz-2.6.18 .. clock=acpi_pm divider=10`
- For Vmware machines, install VMWare tools and set it to synchronize its time with the guest: `vmwaretoolbox`. Alternatvily, you can use Oracle Cluster Time Synchronization Service (ctssd) (metalink document 551704.1)
- Install further packages:

```
# to know distribution and version of Linux (Red Hat Ent. 5.2 used)
cat /etc/issue
```

```
# to know kernel version (and its errata level) (2.6.18-92 or newer)
uname -r

# to list missed packages:
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n' binutils \
compat-libstdc++-33 \
elfutils-libelf \
elfutils-libelf-devel \
gcc \
gcc-c++ \
glibc \
glibc-common \
glibc-devel \
glibc-headers \
ksh \
libaio \
libaio-devel \
libgcc \
libstdc++ \
libstdc++-devel \
make \
sysstat \
unixODBC \
unixODBC-devel

# for missed packages, install them:
rpm -Uvh libaio-devel-0.3.106-3.2.i386.rpm
rpm -Uvh unixODBC*

# Download the appropriate ASMLib RPMs from OTN.
# to know the kernel verion: uname -rm
# In this case we need:
rpm -Uvh oracleasm-support-2.1.3-1.el5.i386.rpm
rpm -Uvh oracleasm-2.6.18-92.el5-2.0.5-1.el5.i686.rpm
rpm -Uvh oracleasmlib-2.0.4-1.el5.i386.rpm


# SELINUX must be disabled
cat /etc/selinux/config | grep SELINUX=
vi /etc/selinux/config
SELINUX=disabled
shutdown -h now -r
```

- Check the hardware requirements

```
# Hardware Requirements (in cluster nodes)
# At least 1.5 GB of physical memory but practically 1.5 is not fine
grep MemTotal /proc/meminfo

# swap space: same as the amount of physical memory
grep SwapTotal /proc/meminfo

# to display swap and memory in one command:
free

# if you don't have enought swap,
#  you can add swap space by creating a temporary swap file.
#  let's say about 500MB:
```

```
dd if=/dev/zero of=tempswap bs=1k count=500000
chmod 600 tempswap
mke2fs tempswap
mkswap tempswap
swapon tempswap

# 1 GB disk space in /tmp
df -h /tmp
# 8 GB of disk space for Oracle software
df

The size of the shared memory should be at least the greater of
MEMORY_MAX_TARGET and MEMORY_TARGET for each Oracle instance on the computer.
To determine the amount of shared memory available, enter the following
command:
# df -h /dev/shm/
```

- Create the required network configuration (rac2 will be created later):
  - Public and Private interface names must be the same for all nodes.
  - This private hostname does not need to be resolvable through DNS and should be entered in the /etc/hosts file.
  - SCAN VIPs must NOT be in the /etc/hosts file, it must be resolved by DNS. But here I've defined it as a single IP address in the "/etc/hosts" file, which is wrong and will cause the cluster verification to fail, but it allows me to complete the install without the presence of a DNS.
  - If you are using a DNS, Oracle recommends that you add lines to the /etc/hosts file on each node, specifying the public IP, VIP and private addresses.
  - If you configured the IP addresses in a DNS server, then, as the root user, change the hosts search order in /etc/nsswitch.conf on all nodes as shown:

    Old: `hosts: files nis dns`

    New: `hosts: dns files nis`
  - Then restart nscd daemon on each node: `/sbin/service nscd restart`

```
# Network names Resolution
# configure /etc/hosts if no domain server is used (both nodes)
vi /etc/hosts
127.0.0.1       localhost.localdomain   localhost
#eth0 - PUBLIC
192.0.2.100 rac1.mydomain.com rac1
192.0.2.101 rac2.mydomain.com rac2
#VIP
192.0.2.102 rac1-vip.mydomain.com rac1-vip
192.0.2.103 rac2-vip.mydomain.com rac2-vip
#eth1 - PRIVATE
172.0.2.100 rac1-priv
172.0.2.101 rac2-priv

# in real production: the follwing should not be there at all
# SCAN: cluster_name-scan.GNS_subdomain_name
192.0.2.104   rac-scan.mydomain.com rac-scan
```

- Create and configure the required OS users and groups

Note: userid and groupid must be the same in all nodes. You can check them by `id oracle` command.

```
# all group and user ids on all the nodes must have identical id
# Grid Infrastructure (GI) and the Oracle RDBMS home will
# be installed using different users:
/usr/sbin/groupadd -g 501 oinstall
/usr/sbin/groupadd -g 502 dba
/usr/sbin/groupadd -g 504 asmadmin
/usr/sbin/groupadd -g 506 asmdba
/usr/sbin/groupadd -g 507 asmoper
/usr/sbin/useradd -u 501 -g oinstall -G asmadmin,asmdba,asmoper grid
/usr/sbin/useradd -u 502 -g oinstall -G dba,asmdba oracle

# set passwords
passwd oracle
passwd grid

# make sure nobody user exists (if not there, create it useradd nobody)
id nobody

# define the env variables for oracle user
vi /home/oracle/.bash_profile
# Oracle Settings
export EDITOR=vi
TMP=/tmp; export TMP
TMPDIR=$TMP; export TMPDIR
ORACLE_HOSTNAME=rac1.mydomain.com; export ORACLE_HOSTNAME
ORACLE_UNQNAME=ron; export ORACLE_UNQNAME
ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
ORACLE_HOME=$ORACLE_BASE/product/11.2.0/db_1; export ORACLE_HOME
ORACLE_SID=RON1; export ORACLE_SID
ORACLE_TERM=xterm; export ORACLE_TERM
PATH=/usr/sbin:$PATH; export PATH
PATH=$ORACLE_HOME/bin:$PATH; export PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export LD_LIBRARY_PATH
CLASSPATH=$ORACLE HOME/JRE:$ORACLE HOME/jlib:$ORACLE HOME/rdbms/jlib; export
CLASSPATH

# shell startup file
vi /etc/profile
if [ $USER = "oracle" ] || [ $USER = "grid" ]; then
 if [ $SHELL = "/bin/ksh" ]; then
  ulimit -p 16384
  ulimit -n 65536
 else
  ulimit -u 16384 -n 65536
 fi
 umask 022
fi


# for C shell
vi /etc/csh.login
if ( $USER = "oracle" || $USER = "grid" ) then
 limit maxproc 16384
 limit descriptors 65536
endif

# define the env variables for oracle user
```

```
vi /home/grid/.bash_profile
# Oracle Settings
export ORACLE_SID=+ASM1
export EDITOR=vi
export ORACLE_HOME=/u01/11.2.0/grid
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export PATH=$ORACLE_HOME/bin:/bin:/usr/bin:/usr/sbin:/usr/local/bin
```

- Configure kernel parameters and shell limits

  Note: If you make a mistake with a parameter setting and your system does not start, then you must start Linux in the single-user runlevel (runlevel 1). At this runlevel, the /etc/sysctl.conf file is not run.

```
# Kernel Parameters
# to tune thme, refer to metalink document 169706.1
# Append the following to the /etc/sysctl.conf file as the root user:
vi /etc/sysctl.conf

# kernel.shmmax not stated in 11g R2 (max: 4G) (169706.1)
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.aio-max-nr = 1048576
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576

# to take immediate effect
/sbin/sysctl -p

# User Shell Limits
# memlock is used to increase the per-process max locked memory
vi /etc/security/limits.conf
grid soft nproc 2047
grid hard nproc 16384
grid soft nofile 1024
grid hard nofile 65536
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536

vi /etc/pam.d/login
session required pam_limits.so
```

- Create the required directories for the Oracle software:

```
# to know if there is an existing oracle inventory
# from its output, ORACLE_BASE will be parent of oraInventory
more /etc/oraInst.loc
# to identify existing Oracle home directories
more /etc/oratab

# Oracle Inventory Directory
# as a root
mkdir -p /u01/app/oraInventory
chown -R grid:oinstall /u01/app/oraInventory
```

```
chmod -R 775 /u01/app/oraInventory

# Grid Infrastructure Home Directory
mkdir -p /u01/11.2.0/grid
chown -R grid:oinstall /u01/11.2.0/grid
chmod -R 775 /u01/11.2.0/grid

# Oracle Base Directory
mkdir -p /u01/app/oracle
#needed to ensure that dbca is able to run after the rdbms installation
mkdir /u01/app/oracle/cfgtoollogs
chown -R oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/app/oracle

# Oracle RDBMS Home Directory
mkdir -p /u01/app/oracle/product/11.2.0/db_1
chown -R oracle:oinstall /u01/app/oracle/product/11.2.0/db_1
chmod -R 775 /u01/app/oracle/product/11.2.0/db_1
```

- Disable screensavers after logging as root, oracle and grid:
  - In Oracle Linux: Applications-> Preferences-> Screen Saver
- In rac1, partition the disks:

Note: On a real life storage, you would create a single whole-disk partition with exactly 1 MB offset on each LUN to be used as ASM Disk. In fdisk: u (to change units from cylinder to sectors), n, p, 1, 2048, w.

```
# as a root, for the disks /dev/sdb .. /dev/sdd
# confirm they are seen:
ls /dev/sd*
#partition the disks:
fdisk /dev/sdb
# answers: "n", "p", "1", "Return", "Return", "p" and "w"
Note: if the following message appears after the "w" command:
 WARNING: Re-reading the partition table failed with error 16: Device or
 resource  busy, then you can avoid restarting the machine by the following
command: partprobe

# to make sure partions are created
ls -lX /dev/sd*
```

- Shutdown rac1 and the Vmware machine soiftware. Then edit its VMware file of rac1 (with vmx extensions) and add the following entry to allow sharing the disks (make sure the scsi controller number is the one you used):

```
disk.locking = "FALSE"
diskLib.dataCacheMaxSize = "0"
diskLib.dataCacheMaxReadAheadSize = "0"
diskLib.dataCacheMinReadAheadSize = "0"
diskLib.dataCachePageSize = "4096"

scsi1.sharedBus = "virtual"

scsi1:1.deviceType = "disk"
scsi1:2.deviceType = "disk"
scsi1:3.deviceType = "disk"
```

- Copy the folder containing rac1 into a new folder in the same directory structure level. Let's name it "rac2". This will be the second node in the cluster.
- Edit the VMware file of rac2 and edit the following: displayName = "rac2"

- Open rac2, then perform:
  - in a terminal issue: `system-config-network-gui`
  - Remove the devices with the "%.bak" nicknames. To do this, highlight a device, deactivate, then delete it.
  - Highlight the "eth0" interface and click the "Edit" button. Change its IP addresses and gate way: IP 192.0.2.101. Click on the "Hardware Device" tab and click the "Probe" button.
  - For eth1 set its ip address to 172.0.2.101. Do not define a gateway.
  - In DNS tab, change hostname to rac2.mydomain.com.
  - Activate the network cards.
- In rac2, perform:

```
# change the variable in the file
vi /home/oracle/.bash_profile
# mark the following:
# ORACLE_SID=RON1; export ORACLE_SID
ORACLE_HOSTNAME=rac2.localdomain; export ORACLE_HOSTNAME

vi /home/grid/.bash_profile
export ORACLE_SID=+ASM2
```

- Start rac1. Make sure the machines can see each other:

```
ping -c 3 rac1
ping -c 3 rac1-priv
ping -c 3 rac2
ping -c 3 rac2-priv
```

- In rac1, configure ASM drivers:

  Note: If you see that the shared disks are not synced between rac1 and rac2, one of the things you can examine is to see if there is any "debug" command in any of the nodes' vmx files. If you find one, shutdown the node, remove the command from the vmx file and restart.

```
# as root
oracleasm configure -i

Default user to own the driver interface []: grid
Default group to own the driver interface []: oinstall
Start Oracle ASM library driver on boot (y/n) [n]: y
Fix permissions of Oracle ASM disks on boot (y/n) [y]: y
# In all nodes: Load the kernel module using the following command:
/usr/sbin/oracleasm init
# If you have any problems, make sure you have the correct
#  version of the driver:
/usr/sbin/oracleasm update-driver

# mark the shared disks: (one node)
/usr/sbin/oracleasm createdisk DISK1 /dev/sdb1
/usr/sbin/oracleasm createdisk DISK2 /dev/sdc1
/usr/sbin/oracleasm createdisk DISK3 /dev/sdd1

# check the disks are marked and seen:
/usr/sbin/oracleasm listdisks
```

```
# in other nodes:
/usr/sbin/oracleasm scandisks
/usr/sbin/oracleasm listdisks

#If you need to unmark a disk that was used in a createdisk command:
/usr/sbin/oracleasm deletedisk DISK1
/usr/sbin/oracleasm deletedisk DISK2
/usr/sbin/oracleasm deletedisk DISK3
```

2. Oracle Grid Infrastructure installation

```
# in rac1: copy the software in a staging folder
mkdir -p /u01/stage/ora11gr2gridinfra
chown -R grid:oinstall /u01/stage/ora11gr2gridinfra
chmod -R 775 /u01/stage/ora11gr2gridinfra

mkdir -p /u01/stage/ora11gr2db
chown -R oracle:oinstall /u01/stage/ora11gr2db
chmod -R 775 /u01/stage/ora11gr2db

# do not use cluvfy because SSH was not configured.


# if you are installing 11.2.0.2: install the package cvuqdisk-1.0.9-1
su -
cd /u01/stage/ora11gr2gridinfra/stage/cvu/cv/remenv/
rpm -iv cvuqdisk-1.0.9-1.rpm
# install it in rac2 as well


# lunch OUI from the clusterware ( as grid from rac1)
# if logged in in Genome using another user, log out and log in as grid
cd /u01/app/stage/ora11gr2gridinfra
./runInstaller

Installation Option
>Select radio button 'Install and Configure Grid Infrastructure for a Cluster'
>Next

Installation Type
>Select 'Advanced Installation'
>Next

Product Language
>Accept 'English' as language'
>Next

Grid Plug and Play
>cluster name: rac
>SCAN name:rac-scan.mydomain.com
>Make sure 'Configure GNS' is NOT selected
>Next

Cluster Node Information
>Add button
>Hostname:rac2.mydomain.com
>Virtual IP Name: rac2-vip.mydomain.com
>OK
```

```
>"SSH Connectivity" button
>Enter the password
>Setup button
>Test button
>Next

Network Interface Usage
>check the public and private networks are specified correctly
>Next

Storage Option
>Select 'Automatic Storage Management (ASM)'
>Next

Creat ASM Disk Group
>Disk Group Name: DGOCRVOTE (3GB disk: Disk1)
>Redundancy: external
>Next
NOTE: If you see an empty screen for you candidate disks it is likely that
ASMLib has not been properly configured. Try reconfigure them.
If you are sure that ASMLib has been properly configured click on 'Change
Discovery Path' and provide the correct destination.

ASM Password
>Specify and conform the password you want to use
>Next

Failure Isolation Support
>Select NOT to use IPMI
>Next

Privileged OS Groups
>Assign the correct OS groups for OS authentication (mostly default is OK)
>Next

Installation Location
>ORACLE_BASE: /u01/app/oracle
 Software location: /u01/11.2.0/grid
>Next

Create Inventory
>Specify the locations: /u01/app/oraInventory
>Next

Perform Prerequisite Checks
>OUI performs certain checks
>Check that status of all checks is Succeeded
Note: in this example, NPS error can be ignored
# 11.2.0.2 returns "PRVF-5150: Path ORCL:  is not a valid path on all nodes"
# in this case: verify manually the asm device. If it succeeds, ignore the
# error. If it fails, run /etc/init.d/oracleasm configure -i on all  nodes and
#  set grid user and oinstall as a group (Note ID 1210863.1)

# For the error "PRVF-5636 : The DNS response time for an unreachable node
# exceeded "15000" ms", the response time can be measured by the command:
#  time nslookup rac1
```

```
>Next

Summary
>Finish

/* For 64-bit, before you run root.sh  apply patch# 9974223
(this patch should also be applied on RDBMS) :
export PATH=$PATH:/u01/app/11.2.0/grid/OPatch
opatch version
cd /u01/app/oracle/admin/patches
opatch lsinventory -detail -oh /u01/app/11.2.0/grid
unzip p9974223_112020_Linux-x86-64.zip
pwd
/u01/app/oracle/admin/patches
opatch napply -local -oh /u01/app/11.2.0/grid -id 9974223
opatch lsinventory -detail -oh /u01/app/11.2.0/grid
*/

Execute Configuration Scripts
>Run the scripts as instructed in the screen
Note: The scripts must be run on one node at a time.
/*
If you face problems in running root.sh and you want
 to deconfigure the previous run of root.sh:
/u01/11.2.0/grid/crs/install/rootcrs.pl -verbose -deconfig -force
*/
>OK

/* Note: when I ran root.sh in node2, I faced "ORA-15018: diskgroup cannot
be created". I deconfigured what made by root.sh, issued oracleasm
stop, start and then ran root.sh again */

We expect the verification phase to fail with an error relating to the SCAN,
assuming you are not using DNS.
    INFO: Checking Single Client Access Name (SCAN)...
    INFO: Checking name resolution setup for "rac-scan.localdomain"...
    INFO: ERROR:
    INFO: PRVF-4664 : Found inconsistent name resolution entries for SCAN name
"rac-scan.localdomain"
    INFO: ERROR:
    INFO: PRVF-4657 : Name resolution setup check for "rac-scan.localdomain"
(IP address: 192.168.2.201) failed
    INFO: ERROR:
    INFO: PRVF-4664 : Found inconsistent name resolution entries for SCAN name
"rac-scan.localdomain"
    INFO: Verification of SCAN VIP and Listener setup failed
Provided this is the only error, it is safe to ignore this
>Next

Message: The installation of the Grid Infrastructure was successfull.
>Close


Note: If your OS is SUSE Linux, shutting down on node will result in shutting
the other nodes. To workaround:
#cd /etc/rc3.d
#ln -s /etc/init.d/ohasd K07ohasd
```

3. Oracle Grid Infrastructure Patching

```
at time of this writing (Jan, 2011), no patches applied.
```

4. Checking Oracle Grid Infrastructure Status

```
# Application resources
crsctl stat res -t


# Oracle Cluster Ready Services (CRS)
crsctl check crs


# Oracle Cluster Registry (OCR) and Voting DiskCluster Ready Services (CRS)
ocrcheck


# Oracle High Availability Services autostart
su -
/u01/11.2.0/grid/bin/crsctl config crs


# Oracle Automatic Storage Management (ASM)
su - grid
srvctl config asm -a


# Oracle Listener
srvctl config listener -a


# Single Client Access Name (SCAN)
srvctl config scan
```

5. Oracle Database 11g R2 Software Installation

```
# as oracle
cd /u01/app/stage/ora11gr2db/db
unzip L11gR2_database_1of2.zip >/dev/null
unzip L11gR2_database_2of2.zip >/dev/null

cd database
./runInstaller


Configure Security Updates
>Provide your e-mail address, if you want or leave them blank
>Next

Installation Options
>Select 'Install Database software only'
>Next>

Install Type
in 11.2.0.1:
>Select 'Real Application Clusters database installation', and select all
nodes.
```

```
in in 11.2.0.2:
>select Oracle RAC One Node installation
>Use the 'SSH Connectivity' button to configure/test the passwordless SSH
connectivity.
>Next

Product Languages
>Confirm 'English'
>Next

Database Edition
>'Enterprise Edition' is ticked
Optional: Click on Select Options> Select "Oracle Partitioning" and "Oracle
Real Application Testing"
>OK
>Next

Installation Location
>Oracle Base: /u01/app/oracle
Software Location: /u01/app/oracle/product/11.2.0/db_1
>Next

Privileged OS Groups
>OSDBA: dba
>OSOPER: oinstall
>Next

Prerequisite Checks
..OUI performs prerequisite checks
>Check that status of all checks is Succeeded
>If you are sure the unsuccessfull checks can be ignored tick the box 'Ignore
All'
>Next

Summary
>Check summary info
>Finish

Install Product
..OUI installs the db software
>as a root, run the root.sh script on the first node then the other nodes (One
at a time)
>OK

Finish
>Close
```

6. Oracle Database 11g R2 Software Patching

- Patch 9004119 should be applied on 11.2.0.1. Do not apply it to 11.2.0.2. The patch just installes the following scripts which are used for RAC One Node:
    - raconefix — Fixes metadata after an Omotion failure or failover
    - raconeinit — Initialize the database to RAC One Node
    - raconestatus — Check the status of RAC One Node database
    - racone2rac — Upgrade RAC One Node database to RAC
    - Omotion — Migrate database online from one node to another

```
# at time of this writing, Patch #9004119 should be applied for RAC One Node

/* Download and apply patch 9004119 */
-- the patch will be applied in a rolling forward style
su - oracle
export PATH=$PATH:$ORACLE_HOME/OPatch
unzip p9004119_112010_LINUX.zip
cd 9004119

# Verify the OUI Inventory is accessible by OPatch:
opatch lsinventory


# apply the patch
opatch apply
# NOTE: the patch asks you to shutdown the instance before applying the patch.
In this case, you do not need to do so because the patch just installs some
scripts. Proceed with applying the script without stopping the instances in
neither rac1 nor rac2.

# make sure the scripts were installed in both nodes:
cd $ORACLE_HOME/bin
ls racone* Omotion
```

7. Install EM Agent in cluster nodes (if required)

8. ASM Diskgroups Creation

Note: It is Oracle's Best Practise to have an OCR mirror stored in a second disk group. To follow this recommendation add an OCR mirror. Mind that you can only have one OCR in a diskgroup. To add OCR mirror to an Oracle ASM disk group, ensure that the Oracle Clusterware stack is running and

```
ocrconfig -add +ORADATA
```

```
ocrcheck
```

```
# as grid user: start the ASM Configuration Assistant (ASMCA)
#su - grid
cd /u01/11.2.0/grid/bin
./asmca

>Disk Groups tab
>Create button
>Disk Group Name: DGDATA
>Redundancy: External
>Disk2
>OK

>Create button
>Disk Group Name: DGFRA
>Redundancy: External
>Disk3
>OK

>Exit
>Yes
```

9. RAC Database Creation

- After the database is created, create a service.

```
# as oracle
cd /u01/app/oracle/product/11.2.0/db_1/bin
./dbca

Welcome
11.2.0.1:
Select 'Oracle Real Application Clusters database'
11.2.0.2:
Select 'Oracle RAC One Node Database'
>Next

Operations
> choose option 'Create a Database'
>Next

Database Template
>Select General Purpose or any template
>Next

Database Identification
>Configuration Type: Admin
>Globale Database Name: ron
>SID: ron
11.2.0.2:
>Serive Name: ronsrv
11.2.0.1:
> Select the nodes: rac1. Make sure you select just ONE NODE.
11.2.0.2:
> Select the nodes: rac1, rac2
>Next

Management Options
>Select the option you want. I selected "Configure Enterprise Manager"
>Next

Database Credentials
>Set the password(s)
>Next

Database File Locations
>Oracle-Managed Files
>Database Area: +DGDATA
>Practically (but not in this case), you should define 'Multiplex Redo Logs
and Control Files'.
>Next

/* Note: If you cannot see the diskgroups, perform the following (ID:
1177483.1):
   su -
   cd <Grid_Home>/bin
   chmod 6751 oracle
   ls -l oracle
   -rwsr-s--x 1 grid oinstall
*/
```

```
ASM Credentials
..If you chose to set up EM, you will be asked about ASMSNMP password
>Enter the password
>Ok button

Recovery Configuration
>Flash recovery area: +DGFRA
>define the size: 2000 MB
 If the size is smaller than recommended a warning will popup.
>Next

Database Content
>Select if you want to have sample schemas created in your database
>Next

Initialization Parameters
>Review and change the settings for memory allocation, characterset etc.
>Next

Database Storage
>Review the database storage settings and change as required
>Next

Creation Options
>Make sure the tickbox 'Create Database' is ticked
>Finish

Summary
>OK
.. Database creation proceeding
>after completion Exit


/* Fix /home/oracle/.bash_profile */
vi /home/oracle/.bash_profile
ORACLE_SID=RON_1


/*  Check Confirmation */
# to show the current configuration and status of the RAC database
srvctl config database -d ron
srvctl status database -d ron

sqlplus /nolog
conn system@ron
select instance_name, status from v$instance ;
select name, db_unique_name from v$database;


# check OEM (if configured):
https://rac1.mydomain.com:1158/em/
# if not started, you can start it:
su - oracle
cd /u01/app/oracle/product/11.2.0/db_1/bin
export ORACLE_UNQNAME=ron
export ORACLE_SID=RON_1
```

```
./emctl status dbconsole
./emctl start dbconsole


# check Oracle processes:
ps -eo pid -o command | grep ora_ | grep -v grep


/* In 11.2.0.1: Create the db service(s) */
OEM home-> Availability-> Cluster Managed Database Services-> enter
credentials-> Create Service-> enter Service Name: hr_srv-> Ok button

# check service status:
srvctl config service -d ron
```

10. Initialize the Database to RAC One Node (11.2.0.1 Only)

- The raconeinit utility renames the db instance and creates the directories and files supporting the renamed instance.
- Renaming the instance name resulted in the DB Control to return the error: "ORA-12505: TNS:listener does not currently know of SID given in connect descriptor". Personally, the only solution I found was to recreate the DB Control configuration as follows. Refer to the "Administrator's Guide" for further details about administering DB Control.

```
srvctl status database -d ron
raconestatus
raconeinit
Output>>
Candidate Databases on this cluster:
 #      Database        RAC One Node    Fix Required
===     ========        ============    ============
[1]          ron             NO              N/A
Enter the database to initialize [1]:
Database ron is now running on server rac1
Candidate servers that may be used for this DB:   rac2
Enter the names of additional candidate servers where this DB may run (space
delimited): rac2
Please wait, this may take a few minutes to finish.......
Database configuration modified.


# check the new instance was created:
srvctl status database -d ron
srvctl config database -d ron


# if DB Control returns ORA-12505, you can recreate its repository as follows:
/* Drop the DB Config Files and Rep Objects */
# Remove the following directories from your file system:
# $ORACLE_HOME/hostname_sid
# $ORACLE_HOME/oc4j/j2ee/OC4J_DBConsole_hostname_sid
rm -r $ORACLE_HOME/rac1_ron
rm -r $ORACLE_HOME/oc4j/j2ee/OC4J_DBConsole_rac1_ron

# $ORACLE_HOME/sysman/admin/emdrep/bin/RepManager hostname listener_port sid -
action drop
$ORACLE_HOME/sysman/admin/emdrep/bin/RepManager rac1 1521 ron_1 -action drop
```

```
/* Add DB Control Config */
[oracle@rac1 ~]$ $ORACLE_HOME/bin/emca -config dbcontrol db -repos create -
cluster

STARTED EMCA at Jan 4, 2011 12:18:02 PM
EM Configuration Assistant, Version 11.2.0.0.2 Production
Copyright (c) 2003, 2005, Oracle.  All rights reserved.

Enter the following information:
Database unique name: ron
Service name: ron
Listener port number: 1521
Listener ORACLE_HOME [ /u01/11.2.0/grid ]: /u01/11.2.0/grid
Password for SYS user:
Password for DBSNMP user:
Password for SYSMAN user:
Cluster name: ron
Email address for notifications (optional):
Outgoing Mail (SMTP) server for notifications (optional):
ASM ORACLE_HOME [ /u01/11.2.0/grid ]: /u01/11.2.0/grid
ASM port [ 1521 ]:
ASM username [ ASMSNMP ]:
ASM user password:
```

11. Postinstallation tasks

```
# backup the root.sh script (on all nodes)
cp /u01/app/oracle/product/11.2.0/db_1/root.sh ~/root.sh.bak
```

12. General Useful Postinstallation Tasks in Linux

- Following are tips to consider after the successful installation to make managing RAC easier.
- Consider using rlwrap utility with SQL*Plus and RMAN:
    - Using rlwrap Utility with RMAN in Unix-Based Systems
    - Using rlwrap Utility with SQL*Plus in Unix-Based Systems

```
/* Easy Acces to crs and db homes */
# it is common to access bin directories in clusterware and db homes

# add the following to .bashrc of oracle user
alias db='cd /u01/app/oracle/product/11.2.0/db_1/bin'
```

Instance Relocation using Omotion (11.2.0.1)

- "Omotion" moves one instance from one node to another in a cluster.
- To migrate an instance without client interruption, implement FAN. You can then also enable TAF.
- You can also provide the database name and migration time as inputs on the command line:

```
./Omotion -d adm2 -m 5
```

```
# in client tnsnames.ora implement TAF:
RONTAF =
 (DESCRIPTION =
   (ENABLE = BROKEN)
   (LOAD_BALANCE = OFF)
   (FAILOVER = ON)
   (ADDRESS = (PROTOCOL = TCP)(HOST = rac-scan.mydomain.com)(PORT = 1521))
     (CONNECT_DATA =
     (SERVER = DEDICATED)
     (SERVICE_NAME = ron)
     (FAILOVER_MODE =
       (TYPE = SELECT)
       (METHOD = BASIC)
       (BACKUP = RON)
     )
   )
 )


# check on which instance the db is running:
srvctl status database -d ron


# to test the TAF, issue a long query in a client


# migrate the instance (-v verbose):
./Omotion -v
```

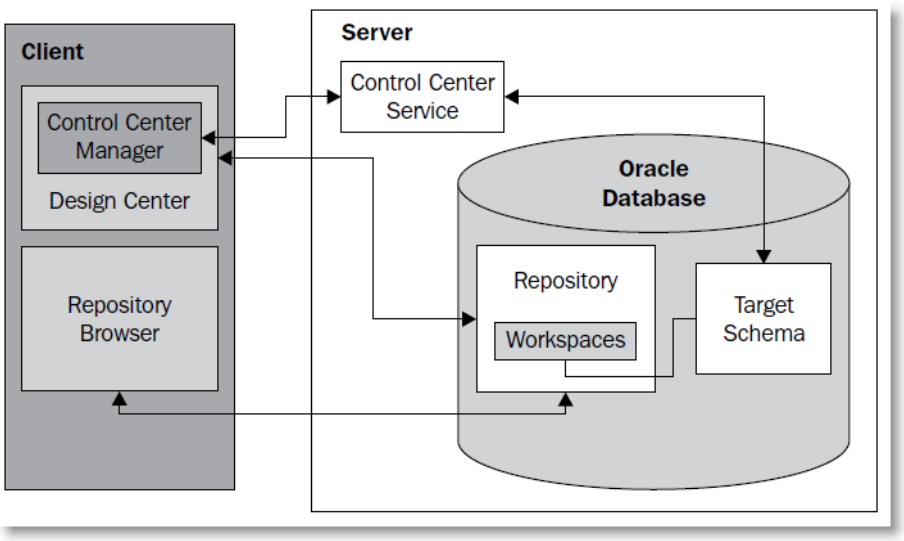
Instance Relocation using Omotion (11.2.0.2)

```
srvctl relocate database -d rac1 -n node7
Coninue>>
```

Part 9 Oracle Warehousing

Oracle Warehouse Builder (OWB)

Oracle Warehouse Builder Architecture



Starting and Stoping the Service

```
Starting and Stopping OWB Service

-- stop service
sqlplus OWBSYS/<password> @<ORACLE_HOME>\stop_service.sql

-- start service
sqlplus OWBSYS/<password> @<ORACLE_HOME>\start_service.sql

-- if start fails, check the problem using the following script:
sqlplus OWBSYS/<password> @<ORACLE_HOME>\service_doctor.sql

-- if OWBSYS user is locked or its password was modified, the start
--  will fail. Store the changed password in the repository using:
sqlplus OWBSYS/<password> @<ORACLE_HOME>\set_repository_password.sql
```

Configuring the Repository and Workspaces

- Goto Start>Oracle Home> Warehouse Builder>Administration>Repository Assistant

  on UNIX, locate OWB_ORACLE_HOME/owb/bin/ and run reposinst.sh
- Select Manage Workspaces> Create a New Workspace> Workspace Owner> ...

  The user must have `OWB_USER` role.
- Select a base language for the repository (cannot be changed later)
- Select Workspace users

Steps of Using Warehouse Builder

- Design the target structure
  - Design a dimensional model and implement it in a database as:
    - Relational implementation (star schema)
    - Multidimensional implementation (OLAP)

- Create a project

  Tip: to delete a project, expand another project then right click on the project you want to delete.

- Create source module(s):
  - Oracle module: specify hostname:port rather than using SQL*Net Connection.
  - Non-Oracle module: create ODBC connection to the database then configure Oracle to connect to the ODBC connection:
    - Copy the file ORACLE_HOME\hs\admin\initdg4odbc.ora to init<odbc connection>.ora

      ```
      HS_FDS_CONNECT_INFO = <odbc data_source_name>
      HS_FDS_TRACE_LEVEL = <trace_level>
      ```

    - For example:

      ```
      HS_FDS_CONNECT_INFO = MYSQLSERVER
      HS_FDS_TRACE_LEVEL = 0
      ```

    - Edit the listener.ora then restart the listener:

      ```
      # PROGRAM=dg4odbc in Oracle 11g
      # PROGRAM=hsodbc in 10g
      SID_LIST_LISTENER=
       (SID_LIST=
        (S Global Explorer> Secutiryt>Users >NewID_DESC=
         (SID_NAME=mysqlserver)
         (ORACLE_HOME=D:\app\bob\product\11.1.0\db_1)
         (PROGRAM=dg4odbc)
       )
      )
      ```

    - Right-click on the ODBC node in the Design Center and select New
    - Edit the connection. Make sure that both username and password are enclosed in double quotes. Enter the Host where the Oracle database resides.

    Tip: to **delete** a connection, unregister it first from Control Manager then delete it from Design Center.

    Note: The *transparent gateway agents* must be purchased and installed separately from the Oracle Database, and then configured to support the communication with the external database. The *generic connectivity agent* is included with the Oracle Database.

  - Import source metadata from files:
    - Files> New> define path (not file)
    - On the created object> Import..> Sample>.. >Finish

- Import source metadata: by using the Import Metadata Wizard or manually. Manual method: under Tables node, New> Data Object Editor.

- Create the target user

  Global Explorer> Secutiryt>Users >New (always created in the db repository)

- Create the target module

- Create target dimensions in OWB:
  - Dimension: Dimension Attributes, Levels, Level Attributes and Hierarchies
  - Time dimension: Project Explorer> Databases> Ourproject> Dimensions> New> Using Time Wizard. A mapping and a sequence will be as a result created.

- Create target cubes in OWB

- Design staging area table(s)

- Create a mapping from source to staging table: add source tables, add the staging target table then link from source to staging target. Linking may include:
  - Joiner: edit it to add more Input Groups, link source tables to it, define join condition. **Note:** Oracle 10.2.0.4 returns "Bad expression return type" error (bug ID 7417869).
  - Aggregator: link output of Joiner to its input. Set its Group By Clause setting. Add required aggregation functions (usually SUM) by right clicking on OUTGRP1> Open details> Output Attributes> Add button to add something like AMOUNT> OK> click on AMOUNT in the aggregator> Properties Window> Expression> set the function> OK
  - Transformation

- Validate the mapping

- Generate the mapping

- Deploy the mapping

- Execute the mapping

Mapping Operators

- **Source and Target Operators**: Cube, Dimension, External Table, Table, Constant, View, Sequence, Construct (returns SYS_REFCURSOR)
- **Data Flow Operators**: Aggregator, Deduplicator (distinct SQL function), Expression, Filter, Joiner, Key Lookup, Pivot, Set Operation, Splitter, Transformation, Table Function
- **Pre/Post Processing Operators**: Mapping Input Parameter, Mapping Output Parameter, Post-Mapping Process, Pre-Mapping Process
- **Pluggable Mappings**: group of operators act as single operator

Part 10 Oracle Database Utilities

Using SQL*Plus

Using SQL*Plus Command-Line Options

`sqlplus [ [<option>] [<logon>] [<start>] ]`

- `<start> is: @<URL>|<filename>[.<ext>] [<parameter> ...]`
- The silent option (-S): no output on screen
- The no-prompt logon option (-L): no username/password question after login fail
- The markup option (-M): HTML generation

```
sqlplus –help
```

Starting SQL*Plus Session

```
$export ORACLE_HOME= /u01/app/oracle/product/11.1.0/db_1
$ORACLE_SID=orcl
$export ORACLE_SID
NLS_LANG and ORA_NLS11
sqlplus oe/oe@//myserver.mydomain:1521/testdb
sqlplus username/passwd
sqlplus username/passwd AS SYSDBA
sqlplus username/passwd@(DESCRIPTION =(ADDRESS=(PROTOCOL=tcp)(HOST=sales-
server)(PORT=1521)(CONNECT_DATA= (SERVICE_NAME=orcl.mycompany.com)))
```

Controlling User Privileges in SQL*Plus

- Product_user_profile table controls what commands a user can issue in SQL*Plus session.
- The following commands can be prevented by using this feature.

  SQL: ALTER, AUDIT, ANALYZE, CREATE, DELETE, DROP, INSERT, LOCK, NOAUDIT, RENAME, SELECT, UPDATE, VALIDATE, TRUNCATE, GRANT, REVOKE, SET ROLE, SET TRANSACTION

  PL/SQL:DECLARE, BEGIN

  SQL*PLUS: COPY, HOST, SET, EDIT, PASSWORD, SPOOL, EXECUTE, QUIT, START, EXIT, RUN, GET, SAVE

- You can use the RESTRICT command to prevent users from using certain operating system commands. It disables the commands even where there are no connections to the server.

Command	Level 1	Level 2	Level 3
EDIT	Disabled	Disabled	Disabled
GET			Disabled
HOST	Disabled	Disabled	Disabled
SAVE		Disabled	Disabled
SPOOL		Disabled	Disabled
START			Disabled
STORE		Disabled	Disabled

```
SQL> DESC product_user_profile
Name    Null?   Type
-----------------------------
PRODUCT         NOT NULL        VARCHAR2(30)
USERID                          VARCHAR2(30)
ATTRIBUTE                       VARCHAR2(240) -- command to disable in upper
case
SCOPE                           VARCHAR2(240)
NUMERIC_VALUE                   NUMBER(15,2)
CHAR_VALUE                      VARCHAR2(240) -- role name to disable
DATE_VALUE                      DATE
LONG_VALUE                      LONG

-- no insert
INSERT INTO product_user_profile VALUES
('SQL*PLUS','OE','INSERT',NULL,NULL,NULL,NULL,NULL);
-- no OS command
INSERT INTO product_user_profile (product,userid,attribute) VALUES
('SQL*Plus','myuser','HOST');
-- no set Role to DBA
insert into product_user_profile(product, userid, attribute, char_value)
values('SQL*Plus', 'APPS', 'ROLES', 'DBA');
-- Preventing access using PL/SQL
insert into system.product_profile (product, userid, attribute, char_value)
values ('SQL*Plus', 'AMAR', 'DECLARE', 'DISABLED');
insert into system.product_profile (product, userid, attribute, char_value)
values ('SQL*Plus', 'AMAR', 'BEGIN', 'DISABLED');

sqlplus -RESTRICT 1
```

Setting the SQL*Plus Environment with the SET Command

help index list of available commands by typing

help set variable possible values

AUTO[COMMIT] Specifies whether commits of transactions are automatic or manual.

DEF[INE]{&/C/ON/OFF} Sets the prefix character used during variable substitutions.

ECHO {OFF/ON} when ON, each command will be displayed before its output onscreen.

FEED[BACK] {OFF/ON} whether to show the number of records returned by your query.

LONG {80/n} maximum width of the LONG, CLOB, NCLOB, and XMLType values

NEWP[AGE] {1/n/none} Specifies the number of blank lines at the top of each new page.

PAGES[IZE] {24/n} Specifies the number of lines in each page

TI[ME] {OFF/ON} Displays time if set to on.

TIMI[NG] {OFF/ON} Controls the display of timing for SQL commands.

VER[IFY] {OFF/ON} Specifies whether SQL text is displayed after variable substitution.

Setting SQL*Plus Preferances

- Globally glogin.sql
- User level login.sql: in Oracle 11g, executed even after connect command

Logging SQL*Plus Errors

- In Oracle 11g, SET ERRORLOGGING ON command writes all subsequent errors into a table called sperrorlog.

```
SET ERRORLOGGING ON
SHOW ERRORLOGGING
```

Key SQL*Plus "Working" Commands

```
Setting SQLPROMPT Command
_USER
_CONNECT_IDENTIFIER
_PRIVILEGE  as sysdba
_DATE
SET SQLPROMPT "_USER > "

host
HOST cp /u01/app/oracle/new.sql /tmp

SPOOL { file_name[.ext] [CRE[ATE]|REP[LACE]|APP[END]]| OFF}
SPOOL /u01/app/oracle/data/employees.txt;
SELECT * FROM hr.employees;
SPOOL OFF;

-- ACCEPT and PROMPT Commands
ACCEPT MYTITLE PROMPT 'Title: '

-- using PAUSE to check long listing output
SQL> SHOW PAUSE
PAUSE is OFF
SQL> SET PAUSE ON
SQL> very long list is displayed
```

Creating Command Files in SQL*Plus

```
SAVE status.sql
-- set the search directory using env var ORACLE_PATH
@myscriptfile.sql

-- Creating a Windows Batch Script
sqlplus username/password@connect_identifier @C:\temp\testscript.sql

-- substitution vars
DEFINE owner = '&1'
SELECT segment_name,segment_type,extents
 FROM dba_segments
 WHERE owner = upper ('&owner');
```

Copying Tables with the COPY Command

- commits while data are moved
- slower than CREATE TABLE .. AS SELECT

```
SQL> COPY
usage: COPY FROM <db> TO <db> <opt> <table> { (<cols>) } USING <sel>
<db> : database string, e.g., hr/your_password@d:chicago-mktg
<opt> : ONE of the keywords: APPEND, CREATE, INSERT or REPLACE
<table>: name of the destination table
<cols> : a comma-separated list of destination column aliases
<sel> : any valid SQL SELECT statement

set ARRAYSIZE 100
SQL> COPY FROM sysadm/sysadm1@finance1-
> CREATE test01 -
> USING SELECT * FROM employee;
```

Creating Web Pages Using SQL*Plus

```
SQLPLUS -MARKUP "HTML ON"
SQL> SET MARKUP HTML ON SPOOL ON
SQL> commands here . . .
SQL> SET MARKUP HTML OFF SPOOL OFF
```

Using SQL to Generate SQL

```
SET ECHO OFF HEADING OFF FEEDBACK OFF
SPOOL test.txt
SELECT 'grant connect, resource to '||username||';' FROM dba_users;
@test.txt
```

Enabling AUTOTRACE for a User

```
-- create plan_table (if not there)
desc plan_table
-- login as the user who wants to use the AUTOTRACE
conn / as sysdba
@d:\oracle\product\10.1.0\db_1\RDBMS\ADMIN\utlxplan.sql

-- create plustrace role
@d:\oracle\product\10.1.0\db_1\sqlplus\admin\plustrce.sql

grant plustrace to scott;

-- the following privilege is also required:
grant select any dictionary to test;

SET AUTOTRACE
 ON
 ON STATISTICS
 ON EXPLAIN
 TRACEONLY
 TRACEONLY STATISTICS
 OFF
```

Using rlwrap Utility with SQL*Plus in Unix-Based Systems

- rlwrap utility allows you to navigate history of commands in SQL*Plus and edit them.

```
/* Install */
# Option 1: for rpm version:
# can be downloaded from:
# http://rpm.pbone.net
# http://ivan.kartik.sk
rpm -ivh rlwrap*.rpm


# Option 2: for tar version
# download rlwrap-0.30.tar.gz (search the net or from
http://www.ahmedbaraka.com/download/oracle/rlwrap-0.30.tar.gz )
# unzip the file and install
su -
gunzip rlwrap-0.30.tar.gz
tar -xvf rlwrap-0.30.tar
cd rlwrap-0.30
./configure
make
make install

/* Recommended */
vi /home/oracle/.bashrc
alias sqlpus='rlwrap sqlplus /nolog'
# or
echo "alias sqlpus='rlwrap sqlplus /nolog'" >> /home/oracle/.bashrc
```

Escaping Special Characters

```
/* Escape quotes */
Use two quotes for every one displayed. Examples:
SELECT 'Frank''s Oracle site' AS text FROM DUAL;
 TEXT
 --------------------
 Franks's Oracle site

SELECT 'A ''quoted'' word.' AS text FROM DUAL;
 TEXT
 ----------------
 A 'quoted' word.

 SQL> SELECT 'A ''''double quoted'''' word.' AS text FROM DUAL;
 TEXT
 -------------------------
 A ''double quoted'' word.


/* Escape wildcard characters */

The LIKE keyword allows for string searches. The '_' wild card character is
used to match exactly one character, while '%' is used to match zero or more
occurrences of any characters. These characters can be escaped in SQL.
Examples:
SELECT name FROM emp
```

```
 WHERE id LIKE '%/_%' ESCAPE '/';
SELECT name FROM emp
 WHERE id LIKE '%\%%' ESCAPE '\';
Escape ampersand (&) characters in SQL*Plus

When using SQL*Plus, the DEFINE setting can be changed to allow &'s
(ampersands) to be used in text:

SET DEFINE ~
-- if SP2-0317 returned, try SET DEF &
SELECT 'Laurel & Hardy' FROM dual;
Other methods:

Define an escape character:

SET ESCAPE '\'
SELECT '\&abc' FROM dual;
Don't scan for substitution variables:

SET SCAN OFF
SELECT '&ABC' x FROM dual;
Another way to escape the & would be to use concatenation, which would not
require any SET commands -

SELECT 'Laurel ' || '&' || ' Hardy' FROM dual;

/* Use the 10g Quoting mechanism: */
Syntax
 q'[QUOTE_CHAR]Text[QUOTE_CHAR]'
 Make sure that the QUOTE_CHAR followed by an ' doesn't exist in the text.
SELECT q'{This is Orafaq's 'quoted' text field}' FROM DUAL;
```

Using SQL*Loader Utility

Invoking SQL*Loader

```
SQLLDR keyword=value [,keyword=value,. . .]
```

Parameters can be specified in a parameter file:

```
sqlldr PARFILE=c:\...\myparfile.txt
```

You specify the command-line parameters (which also could be specified in the control file):

```
USERID = usrname/passwored
CONTROL = '/../mycontrol.ctl'
DATA = '/../mydata.dat'
LOG = '/.../mylog.log'
BAD = '/.../mybadfile.bad'          if not specified, Oracle will create one
DISCARD='/../mydiscard.dat'         discarded data do not meet the criteria
DISCARDMAX=100
SKIP = 235550
in conventional path mode, bind size in bytes OR number of rows
 BINDSIZE = 512000
 ROWS = 64000
DIRECT=true
UNRECOVERABLE=Y    (see direct load options in the following sections)
ERRORS = 0         no errors tolerated
LOAD = 10000       maximum number of logical records to be loaded into the table
SILENT = ALL       all generated message are NOT displayed
PARALLEL=true      only when DIRECT=true
RESUMABLE=true     default is false
RESUMABLE_NAME = finance1_load
RESUMABLE_TIMEOUT = 3660  in seconds
```

Using SQL*Loader Control File

- Datafile:
  - Input File
    - External Input file: `INFILE='/a01/app/oracle/oradata/load/consumer.dat'`
    - Inline Data: `INFILE = * then BEGINDATA ...`
  - Physical and Logical Records
    - `CONCATENATE n RECLEN m` every n rows of data is considered one logical row, and the row length is m characters
    - `CONTINUEIF THIS (1:4) = 'next'` if SQL*Loader finds the four letters next at the beginning of a line, it should treat the data that follows as a continuation of the previous line
    - `CONTINUEIF LAST = '&'` continue if last character is &
  - Record Format
    - specific record terminator: by default new line. To change it to something else like the pipe for example:
      `infile "test.dat" "str '|\n'"`

in this case, the following data are two records (multi-line fields):
```
one line;hello dear world;|
two lines;Dear world,
hello!;|
```

- Fixed record format: `INFILE '..' "fix 12"` (12-byte-size records)
  `ahmed,1234, johnso,1234= "ahmed,1234, " and "12johnson,1234"`
- Variable record format: `INFILE '..' "var 2"`
  `05ahmed12johnson,1234 = "ahmed" and "12johnson,1234"`

- Loading:
  - defining destination table(s): `INTO TABLE .. INSERT | REPLACE | APPEND`
  - defining rejection condition: `WHEN <condition>`
    `WHEN(activity_type <>'H') and (activity_type <>'T')`

Note: OR operator cannot be used in the WHEN condition. To workaraound, load into multiple tables. Check the examples.

- Table- and Field-Mapping
  - Fields position (either you use this or the delimiters)
    - Relative: employee_name to start in position 7 and continue for 30 characters
      ```
      employee_id POSITION(*) NUMBER EXTERNAL 6
      employee_name POSITION(*) CHAR 30
      ```
    - Absolute
      ```
      employee_id POSITION(1:6) INTEGER EXTERNAL
      employee_name POSITION(7:36) CHAR
      ```

Note: a field can be skipped using FILLER keyword.

  - Data Types:
    - INTEGER(n)—binary integer, where n can be 1, 2, 4, or 8
    - SMALLINT
    - CHAR
    - INTEGER EXTERNAL
    - FLOAT EXTERNAL
    - DECIMAL EXTERNAL
    - LOBFILE
  - Delimiters: using POSITION is faster than delimiters
    - `TERMINATED BY WHITESPACE`
    - `TERMINATED BY ","`
    - `FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'`
  - Data Transformation Parameters: SQL function(s) is specified after the data type and should be enclosed in double quotation marks
    - `field_name CHAR TERMINATED BY "," "SUBSTR(:field_name, 1, 10)"`
    - `employee_name POSITION 32-62 CHAR "UPPER(:ename)"`
    - `salary position 75 CHAR "TO_NUMBER(:sal,'$99,999.99')"`
    - `commission INTEGER EXTERNAL "":commission * 100"`

- Command-Line Parameters in the Control File

```
USERID = usrname/passwored
CONTROL = '/../mycontrol.ctl'
DATA = '/../mydata.dat'
LOG = '/.../mylog.log'
BAD = '/.../mybadfile.bad'        if not specified, Oracle will create one
DISCARD='/../mydiscard.dat'       discarded data do not meet the criteria
DISCARDMAX=100
```

SKIP = 235550

in conventional path mode, bind size in bytes OR number of rows

BINDSIZE = 512000

ROWS = 64000

DIRECT=true (see direct load options in the following sections)

ERRORS = 0 no errors tolerated

LOAD = 10000 maximum number of logical records to be loaded into the table

SILENT = ALL all generated message are NOT displayed

PARALLEL=true only when DIRECT=true

RESUMABLE=true default is false

RESUMABLE_NAME = finance1_load

RESUMABLE_TIMEOUT = 3660 in seconds

- Generating Data
  - Constant
    loaded_by CONSTANT "sysadm"
  - Expression: SQL or PL/SQL function (doesn't work in direct loading)
    column_name EXPRESSION "SQL string"
  - Record Number in the datafile
    record_num RECNUM
  - System Date
    loaded_date sysdate
  - Sequence
    loadseq SEQUENCE(max,1)

```
/* Example 1 */
-- tabe separated with nulls in the data
Data to Import:
1       Ahmed Baraka    1000    1.87    1-1-2000
2       John Rice       5000    2.4     10-5-1998
3       Emme Rak        2500    2.34
4       King Size       2700
5       Small Size      3000            31-3-2001


Table Structure
PERSONS
(ID NUMBER,
 PNAME VARCHAR2(100),
 BALANCE NUMBER,
 RATE NUMBER,
 JOIN_DATE DATE );

Control File:
OPTIONS ( ERRORS=0)
LOAD DATA
INFILE 'C:\temp\data\persons.dat'
BADFILE 'C:\temp\data\persons.bad'
DISCARDFILE 'C:\temp\data\persons.dsc'

INTO TABLE "HR"."PERSONS" REPLACE
FIELDS TERMINATED BY X'9' TRAILING NULLCOLS
```

```
(ID INTEGER EXTERNAL,
 PNAME CHAR,
 BALANCE INTEGER EXTERNAL,
 RATE FLOAT EXTERNAL,
 JOIN_DATE date 'dd-mm-yyyy')



/* Example 2 */
-- positional columns
load data
 infile *
 replace
 into table departments
 (  dept     position (02:05) char(4),
    deptname position (08:27) char(20)
 )
begindata
COSC  COMPUTER SCIENCE
ENGL  ENGLISH LITERATURE
MATH  MATHEMATICS
POLY  POLITICAL SCIENCE

/* Example 3 */
-- data transformation
LOAD DATA
  INFILE *
  INTO TABLE modified_data
  (  rec_no                    "my_db_sequence.nextval",
     region                    CONSTANT '31',
     time_loaded               "to_char(SYSDATE, 'HH24:MI')",
     data1        POSITION(1:5)  ":data1/100",
     data2        POSITION(6:15) "upper(:data2)",
     data3        POSITION(16:22)"to_date(:data3, 'YYMMDD')"
  )
BEGINDATA
11111AAAAAAAAAA991201
22222BBBBBBBBBB990112

LOAD DATA
  INFILE 'mail_orders.txt'
  BADFILE 'bad_orders.txt'
  APPEND
  INTO TABLE mailing_list
  FIELDS TERMINATED BY ","
  (  addr,
     city,
     state,
     zipcode,
     mailing_addr   "decode(:mailing_addr, null, :addr, :mailing_addr)",
     mailing_city   "decode(:mailing_city, null, :city, :mailing_city)",
     mailing_state,
     move_date      "substr(:move_date, 3, 2) || substr(:move_date, 7, 2)"
  )


/* Example 4 */
```

```
-- loading from multiple input files
LOAD DATA
  INFILE file1.dat
  INFILE file2.dat
  INFILE file3.dat
  APPEND
  INTO TABLE emp
  ( empno  POSITION(1:4)   INTEGER EXTERNAL,
    ename  POSITION(6:15)  CHAR,
    deptno POSITION(17:18) CHAR,
    mgr    POSITION(20:23) INTEGER EXTERNAL
  )


/* Example 5 */
-- loading into multiple tables
-- skipping columns (FILLER)
--  POSITION(1:4) in the example is a must to resent the pointer back
--  to the beginning of the row
--  In delimited formats, use "POSITION(1)" after the first
--  column to reset the pointer
LOAD DATA
 INFILE *
 INTO TABLE tab1 WHEN tab = 'tab1'
   ( tab  FILLER CHAR(4),
     col1 INTEGER
   )
 INTO TABLE tab2 WHEN tab = 'tab2'
   ( tab  FILLER POSITION(1:4),
     col1 INTEGER
   )
BEGINDATA
tab1|1
tab1|2
tab2|2
tab3|3

-- another example
LOAD DATA
 INFILE 'mydata.dat'
 REPLACE
 INTO TABLE emp
      WHEN empno != ' '
 ( empno  POSITION(1:4)   INTEGER EXTERNAL,
   ename  POSITION(6:15)  CHAR,
   deptno POSITION(17:18) CHAR,
   mgr    POSITION(20:23) INTEGER EXTERNAL
 )
 INTO TABLE proj
      WHEN projno != ' '
 (  projno POSITION(25:27) INTEGER EXTERNAL,
    empno  POSITION(1:4)   INTEGER EXTERNAL
 )


/* Example 6 */
-- work around on being unable to use OR in the WHEN condition
LOAD DATA
  INFILE  'mydata.dat' BADFILE  'mydata.bad' DISCARDFILE 'mydata.dis'
```

```
  APPEND
  INTO TABLE my_selective_table
  WHEN (01) <> 'H' and (01) <> 'T'
  (
     region              CONSTANT '31',
     service_key         POSITION(01:11)   INTEGER EXTERNAL,
     call_b_no           POSITION(12:29)   CHAR
  )
  INTO TABLE my_selective_table
  WHEN (30:37) = '20031217'
  (
     region              CONSTANT '31',
     service_key         POSITION(01:11)   INTEGER EXTERNAL,
     call_b_no           POSITION(12:29)   CHAR
  )



/* Example 7 */
-- load records with multi-line fields
-- doesn't work with inline data
 load data
 infile "test.dat" "str '|\n'"
 into test_table
 fields terminated by ';' TRAILING NULLCOLS
 (
   desc, txt
 )

test.dat:
 one line;hello dear world;|
 two lines;Dear world,
 hello!;|


/* Example 8 */
-- loading binary files (word, images, video... etc)
CREATE TABLE image_table (
       image_id   NUMBER(5),
       file_name  VARCHAR2(30),
       image_data BLOB);

Control File:

LOAD DATA
INFILE *
INTO TABLE image_table
REPLACE
FIELDS TERMINATED BY ','
(
 image_id   INTEGER(5),
 file_name  CHAR(30),
 image_data LOBFILE (file_name) TERMINATED BY EOF
)
BEGINDATA
001,image1.gif
002,image2.jpg
```

```
003,image3.jpg


/* Example 9 */
-- using specified characterset

LOAD DATA
CHARACTERSET WE8EBCDIC500
INFILE data.ebc "fix 86 buffers 1024"
BADFILE data.bad'
DISCARDFILE data.dsc'
REPLACE
INTO TABLE temp_data
(
 ...

/* Example 10 */
-- Loading a Sequence Number
LOAD DATA
INFILE '/u01/app/oracle/oradata/load/testload.txt'
INSERT INTO TABLE test123
(test_seq.nextval,. . .)
```

Loading Excel File into a Table using SQL*Loader

1. If any cell data has newline characters (Alt+ENTER), remove them using the following Excel script:

```
' Removing tabs and carriage returns from worksheet cells
Sub CleanUp()
 Dim TheCell As Range
 On Error Resume Next

 For Each TheCell In ActiveSheet.UsedRange
   With TheCell
     If .HasFormula = False Then
       .Value = Application.WorksheetFunction.Clean(.Value)
     End If
   End With
 Next TheCell
End Sub
```

2. Save Excel file as CSV
3. Use SQL*Loader to load from CSV.

Loading Large Fields into a Table

If you try to load any field larger than 255 bytes into a table, even if the table column is defined as VARCHAR2(2000) or a CLOB, "Field in datafile exceeds maximum length" error will be returned. You need to specify the size of the table column in the control file:

```
LOAD DATA
INFILE '/u01/app/oracle/oradata/load/testload.txt'
INSERT INTO TABLE test123
```

```
FIELDS TERMINATED BY ','
(text CHAR(2000))
```

Using Direct Load Options

DIRECT = true

DATA_CACHE: (default 1000) used if you have duplicate data and timestamp values in your data.

ROWS: save data every number of ROWS

UNRECOVERABLE=Y (default N) used with direct=true and disables redo log generation

SKIP_INDEX_MAINTENANCE=true (default false) do not bother maintaining the indexes during the load.

COLUMNARRAYROWS: number of rows loaded before the building of the stream buffer

STREAMSIZE: the size of the stream buffer

MULTITHREADING: (default true) parallel conversion of column arrays to stream buffers and stream buffer

Data Pump Export and Import

Data Pump Components

- **The `DBMS_DATAPUMP` package**: this is the main engine of the Data Pump utilities. It contains procedures that do the export and import actions.
- **The `DBMS_METADATA` package**: this package is used to extract and modify data dictionary metadata.
- **The command-line clients, `expdp` and `impdp`.**

Data Pump Export Interfaces

- Using the Command Line

  ```
  expdp system/manager directory=dpump_dir1 dumpfile=expdat1.dmp
  ```

- Using a Parameter File

  ```
  expdp parfile=myfile.txt
  ```

  paramerer file may contain:

  ```
  SCHEMAS=HR
  DIRECTORY=dpump_dir1
  DUMPFILE=system1.dmp
  SCHEMAS=hr ...
  ```

- Using Interactive-command Data Pump Export

  This mode is enabled by pressing [Ctrl] + [C] during an export operation started with the command-line interface or the parameter file interface.

- Using EM Database Control

  Start the Database Control and go to the Maintenance | Utilities page.

Export Modes Parameters

- `FULL` (requires `EXPORT_FULL_DATABASE` role), `SCHEMAS`, `TABLES`, `TABLESPACES`, `TRANSPORT_TABLESPACES`, and `TRANSPORT_FULL_CHECK`

Required Rrivileges

```
# basic privileges:
grant create session, create table, create procedure to datapump_user;

# if you want to do any of the following:
- to run a full database Export or
- to run a transport_tablespace job or
- to run an Export DataPump job with the TRACE parameter or
- to run an operation that exports a different schema.
grant exp_full_database, imp_full_database to datapump_user;
```

Invoking Export Data Pump Examples

```
# the command requires CREATE ANY DIRECTORY privileges
```

```
CREATE DIRECTORY dpump_dir1 AS '/u01/mydir';
GRANT READ, WRITE ON DIRECTORY dpump_dir1 TO baraka;


# schema mode is the default
expdp baraka/password DIRECTORY=dpump_dir1 dumpfile=testexp01.dmp
LOGFILE=dpump_dir2:mylog.log
# generate filenames based on date and time (sed used to get rid of spaces)
expdp sa/s directory=dpdir dumpfile=sa`date +%d-%m-%y_%k-%M | sed
's/[[:space:]]//'`.dmp logfile=dpdir:sa`date +%d-%m-%y_%k-%M | sed
's/[[:space:]]//'`.log

# tables mode
expdp baraka/password tables=employees DIRECTORY=dpump_dir1
dumpfile=testexp01.dmp
expdp system/password tables=hr.employees ..

# nolog file (by default export.log is generated)
expdp … nologfile=y

# overwrite existing dumpfiles (11g)
expdp … REUSE_DUMPFILES=y

# compression: ALL, DATA_ONLY, METADATA_ONLY, NONE
expdp … COMPRESSION=NONE
```

Export Filtering Parameters

```
/* EXCLUDE & INCLUDE */
EXCLUDE=INDEX
EXCLUDE=PROCEDURE
EXCLUDE=TABLE:"LIKE 'EMP%'"
EXCLUDE=SCHEMA:"='HR'"
INCLUDE=TABLE:"IN ('EMP', 'DEPT')"

# When used in command line, use slashes to escape single and double
quotation:
expdp .. schemas=SCOTT EXCLUDE=TABLE:\"=\'EMP\'\"

/* QUERY */
QUERY=OE.ORDERS: "WHERE order_id > 100000 ORDER BY order_date desc"
```

Export Remapping Parameters

```
/* REMAP_DATA   (11g) */
-- the remapping function shouldn’t have commit or rollback
REMAP_DATA=[schema.]tablename.column_name:[schema.]pkg.function

expdp hr/passwd DIRECTORY=dp_dir DUMPFILE=remap.dmp
TABLES=hr.employees REMAP_DATA=hr.employees.last_name:hr.remap_pckg.modifychar
```

Sampling Export Data

- When you sample a parent table, the child table may contain rows unreferenced by the parent. In such case, impdp will generate "ORA-02298: cannot validate (<foreign key constraint name>)-parent keys not found"

```
SAMPLE=[[schema_name.]table_name:]sample_pct
SAMPLE=10
SAMPLE="HR"."EMPLOYEES":50
```

Export Encryption Parameters

- (11g): To secure the exported dump file, the following new parameters are presented in Oracle 11g Data pump: ENCRYPTION, ENCRYPTION_PASSWORD and ENCRYPTION_ALGORITHM. To enable encryption, you must specify either the ENCRYPTION or ENCRYPTION_PASSWORD parameter, or both.

```
ENCRYPTION = {all | data_only | encrypted_columns_only | metadata_only | none}
ENCRYPTION_ALGORITHM = { AES128 | AES192 | AES256 }
ENCRYPTION_MODE = { DUAL | PASSWORD | TRANSPARENT }
```

```
expdp hr DUMPFILE=dp_dir.hr_enc.dmp JOB_NAME=enc ENCRYPTION=data_only
ENCRYPTION_PASSWORD=mypassword

expdp hr DIRECTORY=dp_dir DUMPFILE=hr_enc.dmp
 ENCRYPTION=all ENCRYPTION_PASSWORD=mypassword
 ENCRYPTION_ALGORITHM=AES256 ENCRYPTION_MODE=dual
```

Export Estimating Parameters

```
ESTIMATE={BLOCKS | STATISTICS}
ESTIMATE_ONLY=y
```

```
expdp system/pswd estimate_only=y
```

Export Network Link Parameter

- You can initiate an export job from your server and have Data Pump export data from a remote database to dump files located on the instance from which you initiate the Data Pump export job.
- READ ONLY DB can still be loaded from.

```
expdp hr/hr DIRECTORY=dpump_dir1 NETWORK_LINK=source_database_link
DUMPFILE=network_export.dmp

-- more detailed steps:
-- scenario: I will take a data pump export from database ORCL
-- and dumpfile will be written to database TIGER
sqlplus sa/a@tiger
create database link orcl.net using 'ORCL';
OR
Create database link orcl.net connect to sa identified by a
using '(DESCRIPTION=(ADDRESS = (PROTOCOL = TCP)(HOST =10.4.x.x) (PORT=1521))
(connect_data=(service_name=orcl)))';

select * from dual@orcl.net;

$expdp arju/a@tiger directory=d schemas=arju dumpfile=arju_dump_from_orcl.dmp
```

```
network_link=orcl.net
```

Export PARALELL Parameter

- You should specify number of dump files equal to the PARALLEL value.

```
expdp system/manager full=y
parallel=4
dumpfile=
DIR1:full1%U.dat,
DIR2:full2%U.dat,
DIR3:full3%U.dat,
DIR4:full4%U.dat
filesize = 2G
```

Import Modes Parameters

- You can perform a Data Pump import in various modes, using the TABLE, SCHEMAS, TABLESPACES, and FULL

File- and Directory-Related Parameters

The Data Pump import utility uses the `PARFILE`, `DIRECTORY`, `DUMPFILE`, `LOGFILE`, and `NOLOGFILE` commands in the same way as the Data Pump export utility.

`SQLFILE`

This parameter enables you to extract the DDL from the export dump file, without importing any data.

```
impdp salapati/sammyy1 DIRECTORY=dpump_dir1 DUMPFILE=finance.dmp
SQLFILE=dpump_dir2:finance.sql
```

`REUSE_DATAFILES`

This parameter tells Data Pump whether it should use existing datafiles for creating tablespaces during an import.

Using TABLE_EXISTS_ACTION Parameter

```
TABLE_EXISTS_ACTION=SKIP,APPEND,TRINCATE,REPLACE
```

Import Filtering Parameters

- Same as export filtering parameters

Import Remapping Parameters

```
/* REMAP_TABLE  (11g) */
-- rename table during export or import
-- it won't work if table already exists
 impdp dumpfile=dp_dir:docs.dmp REMAP_TABLE=hr.docs:docs2 userid=hr/password
impdp dumpfile=dp_dir:docs.dmp REMAP_TABLE=hr.docs.part1:docs3
userid=hr/password
```

```
REMAP_SCHEMA
 impdp system/manager dumpfile=newdump.dmp REMAP_SCHEMA=hr:oe


REMAP_DATAFILE
 Changes the name of the source datafile to the target datafile name in all SQL statements where the
 source datafile is referenced: CREATE TABLESPACE, CREATE LIBRARY, and CREATE DIRECTORY.
 Remapping datafiles is useful when you move databases between platforms that have different file
 naming conventions.
 impdp hr/hr FULL=y DIRECTORY=dpump_dir1 DUMPFILE=db_full.dmp
 REMAP_DATAFILE='DB1$:[HRDATA.PAYROLL]tbs6.f':'/db1/hrdata/payroll/tbs6.f'


REMAP_TABLESPACE
 This parameter enables you to move objects from one tablespace into a different tablespace during an
 import.
impdp hr/hr REMAP_TABLESPACE='example_tbs':'new_tbs' DIRECTORY=dpump_dir1
PARALLEL=2 JOB_NAME=cf1n02 DUMPFILE=employees.dmp NOLOGFILE=Y
```

Ignoring Nondeferred Constraints

- (11g): setting the DATA_OPTIONS parameter to SKIP_CONSTRAINT_ERRORS will cause the import program to skip errors generated by the nondeferred database constraints. In the case of deferred constraints, imports will always be rolled back.

```
impdp Robert/robert DIRECTORY=data_pump_dir DUMPFILE=remap.dmp
tables=ROBERT.NAMES data_options=SKIP_CONSTRAINT_ERRORS
```

Import Network Link Parameter

- Get data from a remote database through db link.
- Read Only DB can still be loaded from.
- When the NETWORK_LINK parameter is used in conjunction with the TABLES parameter, only whole tables can be imported (not partitions of tables).
- If the USERID that is executing the import job has the IMP_FULL_DATABASE role on the target database, then that user must also have the EXP_FULL_DATABASE role on the source database.
- The only types of database links supported by Data Pump Import are: public, fixed-user, and connected-user. Current-user database links are not supported.

```
impdp hr/hr TABLES=employees DIRECTORY=dpump_dir1
NETWORK_LINK=source_database_link EXCLUDE=CONSTRAINT
```

Import Flashback Parameters

- FLASHBACK_TIME and FLASHBACK_SCN enable you to import data consistent as of the flashback time you specify in your import job.

```
impdp system/manager FLASHBACK_TIME="TO_TIMESTAMP('01-06-2009 07:00:00;','DD-
MM-YYYY HH24:MI:SS')"
```

Import PARALELL Parameter

- You should specify number of dump files equal to the PARALLEL value.
- If you received the errors: ORA-39029, ORA-31671, ORA-39078, try increasing STREAMS_POOL_SIZE:
  alter system set STREAMS_POOL_SIZE=100M;

```
impdp system/manager
directory = MYDIR
parallel = 4
dumpfile = full1%U.dat,full2%U.dat,
full3%U.dat,full4%U.dat
```

Monitoring a Data Pump Jobs

```
# data pump jobs
SELECT * FROM dba_datapump_jobs;

# data pump sessoins
SELECT sid, serial#
FROM v$session s, dba_datapump_sessions d
WHERE s.saddr = d.saddr;

# viewing job progress
SELECT opname, target_desc, sofar, totalwork
 FROM v$session_longops;
```

LogMiner

Types of Supplemental Logging

- Unconditional supplemental log group: column values are always logged.

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (FOREIGN KEY) COLUMNS;
```

- Conditional supplemental log group: column values are logged if changed.

Levels of Supplemental Logging

- Database Level: more resourse consuming, least required level is:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;
```

- Table Level:

```
ALTER TABLE HR.EMPLOYEES ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;

ALTER TABLE HR.EMPLOYEES ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;

ALTER TABLE HR.EMPLOYEES ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;
```

- Table-Level User-Defined Supplemental Log Groups
  - User-defined unconditional log groups

```
ALTER TABLE HR.EMPLOYEES ADD SUPPLEMENTAL LOG GROUP emp_parttime
(EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID) ALWAYS;
```

  - User-defined conditional supplemental log groups

```
-- always keyword isn't used:
ALTER TABLE HR.EMPLOYEES ADD SUPPLEMENTAL LOG GROUP emp_fulltime
(EMPLOYEE_ID, LAST_NAME, DEPARTMENT_ID);

ALTER TABLE HR.EMPLOYEES ADD SUPPLEMENTAL LOG GROUP emp_parttime(
DEPARTMENT_ID NO LOG, EMPLOYEE_ID);
```

Note: you must enable at least minimal supplemental logging prior to generating log files which will be analyzed by LogMiner.

Disabling Database-Level Supplemental Logging

```
ALTER DATABASE DROP SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;

ALTER DATABASE DROP SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS;

ALTER DATABASE DROP SUPPLEMENTAL LOG DATA;
```

LogMiner Dictionary Options

- Using the Online Catalog: use it, if you believe no DDL is performed during Analysis.
- Extracting a LogMiner Dictionary to the Redo Log Files: good if DDL might be there during analysis
- Extracting the LogMiner Dictionary to a Flat File: forget this option!

Redo Log File Options

- Automatically: required redo log is automaitcally detected and used by the utility.
- Manually: you provide list of the redo files to analyse.

OPTIONS possible values in DBMS_LOGMNR.START_LOGMNR:

DICT_FROM_ONLINE_CATALOG : Using the Online Catalog

DICT_FROM_REDO_LOGS : Start LogMiner

CONTINUOUS_MINE : Redo Log File Options

COMMITTED_DATA_ONLY : Showing Only Committed Transactions

SKIP_CORRUPTION : Skipping Redo Corruptions

NO_SQL_DELIMITER : Formatting Reconstructed SQL Statements for Reexecution

PRINT_PRETTY_SQL : Formatting the Appearance of Returned Data for Readability

NO_ROWID_IN_STMT : Formatting Reconstructed SQL Statements for Reexecution

DDL_DICT_TRACKING : Tracking DDL Statements in the LogMiner Dictionary (not with DICT_FROM_ONLINE_CATALOG)

Obtaining LogMiner Operational Information

```
-- specified redo log files
--  STATUS: 0: the log file will be processed
 1: the file will be the first redo log file to be processed
 2: the file will not be processed by LogMiner (pruned)
 4: the file is missing
select * from V$LOGMNR_LOGS;

-- optional LogMiner parameters
select * from V$LOGMNR_PARAMETERS;

-- current settings for supplemental logging
select SUPPLEMENTAL_LOG_DATA_FK, SUPPLEMENTAL_LOG_DATA_ALL,
       SUPPLEMENTAL_LOG_DATA_UI, SUPPLEMENTAL_LOG_DATA_MIN
from  V$DATABASE;

select * from  DBA_LOG_GROUPS;

select * from  ALL_LOG_GROUPS;

select * from  USER_LOG_GROUPS;
```

```
select * from  DBA_LOG_GROUP_COLUMNS;

select * from  ALL_LOG_GROUP_COLUMNS;

select * from  USER_LOG_GROUP_COLUMNS;
```

Examples of Using LogMiner

Without Sepecifying the Redo Files

```
Example:
 LogMiner Dictionary: Online Catalog,
 Redo Log File Options: Automatically

-- check Supplemental Logging is enabled
(see the examples above)

-- start LogMiner
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS';
begin
DBMS_LOGMNR.START_LOGMNR(
 STARTTIME => '01-Jan-2003 08:30:00',
 ENDTIME => '01-Jan-2003 08:45:00',  -- future value can be set SYSDATE+5/24
 OPTIONS => DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG +
  DBMS_LOGMNR.CONTINUOUS_MINE +
  DBMS_LOGMNR.COMMITTED_DATA_ONLY +
  DBMS_LOGMNR.PRINT_PRETTY_SQL );
end;
/


-- query V$LOGMNR_CONTENTS
SELECT USERNAME AS usr,(XIDUSN || '.' || XIDSLT || '.' || XIDSQN) as XID,
   SQL_REDO FROM V$LOGMNR_CONTENTS
   WHERE SEG_OWNER IS NULL OR SEG_OWNER NOT IN ('SYS', 'SYSTEM') AND
   TIMESTAMP > '10-jan-2003 15:59:53';


SELECT OPERATION, SQL_REDO, SQL_UNDO
   FROM V$LOGMNR_CONTENTS
   WHERE SEG_OWNER = 'OE' AND SEG_NAME = 'ORDERS' AND
   OPERATION = 'DELETE' AND USERNAME = 'RON';

-- Querying Based on Column Values
SELECT SQL_REDO FROM V$LOGMNR_CONTENTS
   WHERE
   SEG_NAME = 'EMPLOYEES' AND
   SEG_OWNER = 'HR' AND
   OPERATION = 'UPDATE' AND
   DBMS_LOGMNR.MINE_VALUE(REDO_VALUE, 'HR.EMPLOYEES.SALARY')
   DBMS_LOGMNR.MINE_VALUE(UNDO_VALUE, 'HR.EMPLOYEES.SALARY');


-- end the Miner session
EXECUTE DBMS_LOGMNR.END_LOGMNR;
```

Wit Sepecifying the Redo Files

```
Example:
 LogMiner Dictionary: Using the LogMiner Dictionary in the Redo Log Files,
 Redo Log File Options: list of files provided
 Redo log file that contains the end of the dictionary extract must have been
created before the redo log file that you want to analyze, but should be as
recent as possible.
Assume: you want to analyze file no 210


-- check Supplemental Logging is enabled
(see the examples above)


- TO extract the data dictionary to the redo logs (must be done before the
redo to analyze)
EXECUTE sys.DBMS_LOGMNR_D.build( OPTIONS =>
sys.DBMS_LOGMNR_D.store_in_redo_logs);

- Find a redo log file that contains the end of the dictionary extract
SELECT NAME, SEQUENCE#, DICTIONARY_BEGIN d_beg, DICTIONARY_END d_end
   FROM V$ARCHIVED_LOG
   WHERE SEQUENCE# = (SELECT MAX (SEQUENCE#) FROM V$ARCHIVED_LOG
   WHERE DICTIONARY_END = 'YES' and SEQUENCE# <= 210);

- Find the redo log file that contains the start of the data dictionary
extract that matches the end of the dictionary found in the previous step:
SELECT NAME, SEQUENCE#, DICTIONARY_BEGIN d_beg, DICTIONARY_END d_end
   FROM V$ARCHIVED_LOG
   WHERE SEQUENCE# = (SELECT MAX (SEQUENCE#) FROM V$ARCHIVED_LOG
   WHERE DICTIONARY_BEGIN = 'YES' and SEQUENCE# <= 208);

- Specify the list of the redo log files of interest. Order doesn't matter:
EXECUTE DBMS_LOGMNR.ADD_LOGFILE(-
   LOGFILENAME => '/usr/oracle/data/db1arch_1_210_482701534.dbf', -
       OPTIONS => DBMS_LOGMNR.NEW);
EXECUTE DBMS_LOGMNR.ADD_LOGFILE(-
   LOGFILENAME => '/usr/oracle/data/db1arch_1_208_482701534.dbf');
EXECUTE DBMS_LOGMNR.ADD_LOGFILE(-
   LOGFILENAME => '/usr/oracle/data/db1arch_1_207_482701534.dbf');

- Query the V$LOGMNR_LOGS :
SELECT FILENAME AS name, LOW_TIME, HIGH_TIME FROM V$LOGMNR_LOGS;


- Start LogMiner:
EXECUTE DBMS_LOGMNR.START_LOGMNR(-
   OPTIONS => DBMS_LOGMNR.DICT_FROM_REDO_LOGS + -
              DBMS_LOGMNR.COMMITTED_DATA_ONLY + -
              DBMS_LOGMNR.PRINT_PRETTY_SQL);

- Query the V$LOGMNR_CONTENTS:
SELECT USERNAME AS usr, SQL_REDO FROM V$LOGMNR_CONTENTS
   WHERE SEG_OWNER IS NULL OR SEG_OWNER NOT IN ('SYS', 'SYSTEM') AND
   TIMESTAMP > '10-jan-2003 15:59:53';

- to display all the DML statements that were executed as part of the CREATE
```

```
TABLE DDL statement:
SELECT SQL_REDO FROM V$LOGMNR_CONTENTS
   WHERE XIDUSN  = 1 and XIDSLT = 2 and XIDSQN = 1594;

-- end the Miner session
EXECUTE DBMS_LOGMNR.END_LOGMNR;
```

Part 11 Miscellaneous Oracle Database Topics

Managing Oracle Database Control

Configuring and Using the Database Control

- Note ID 1099271.1 “Master Note for Enterprise Manager Configuration Assistant (EMCA) in Single Instance Database Environment” is a good reference.
- In one situation, I faced the error "System error 1060 has occurred." when I issued the command emctl start on Oracle 10g for Windows Server. To resolve it:

  Issue the command and get from its output the Agent process ID. Usin the Process manager, kill the corresponding emAgent.exe process. Then issue emctl start command. If error presists, try killing all the emAgent.exe processes.

```
-- ORACLE_SID should be set
export ORACLE_SID=mydb

-- manually configuring db control
emca -config dbcontrol db

emctl status dbconsole
emctl start dbconsole
emctl stop dbconsole
http://localhost:5500/em
$ORACLE_HOME/install/portlist.ini

isqlplusctl start
```

Implementing EM Database Control Auto Startup

```
# (1) as root create oraemctl file
su -
vi /etc/init.d/oraemctl

#!/bin/bash
#
# oraemctl Starting and stopping Oracle Enterprise Manager Database Control.
# Script is valid for 10g and 11g versions.
#
# chkconfig: 35 80 30
# description: Enterprise Manager DB Control startup script

# Source function library.

. /etc/rc.d/init.d/functions

ORACLE_OWNER="oracle"
ORA_HOME="/u01/app/oracle/product/10.2.0/db_1"

case "$1" in
start)
 echo -n $"Starting Oracle EM DB Console:"
 su - $ORACLE_OWNER -c "$ORA_HOME/bin/emctl start dbconsole"
 echo "OK"
```

```
;;
stop)
 echo -n $"Stopping Oracle EM DB Console:"
 su - $ORACLE_OWNER -c "$ORA_HOME/bin/emctl stop dbconsole"
 echo "OK"
;;
*)
echo $"Usage: $0 {start|stop}"
esac


# (2) change permissions and runlevels
chmod 750 /etc/init.d/oraemctl
chkconfig --add oraemctl --level 0356
```

Dropping and Recreating the Management Repository

```
-- Oracle 11g R2:
cd /u01/app/oracle/product/11.2.0/db_1/sysman/admin/emdrep/bin
./RepManager rac1 1521 ron_1 -sys_password qwer5 -action drop -repos_user
sysman

./RepManager rac1 1521 ron_1 -sys_password qwer5 -action create -repos_user
sysman
```

Installing Oracle 10g R5 (10.2) Enterprise Manager Grid Control for Linux x86

Following are the basic installation steps to install Oracle 10.2.0.5 Enterprise Manager Grid Control on Linux on a new database.

Note: for any installation, you should check the Release Notes documenation before taking any practical step.

Note: Since Enterprise Manager 10g Grid Control Release 4 or higher are patch sets, you need to use the 'Installing Software-Only and Configuring Later' installation method as indicated below. This method is not supported by the interactive OUI.

Installation Environment

- Emulation software: VMWare Workstation 7 for Windows
- OS: Redhat Enterprise Linux 5.2 for x86 32-bit

Required Software

- Oracle Enterprise Manager Grid Control 10.2.0.1 for Linux x86 32-bit
- Oracle Enterprise Manager Grid Control 10.2.0.5 for Linux x86 32-bit
- The interim RDBMS patch# 4329444
- Oracle Database Patch Set Notes 10g Release 1 (10.1.0.5) Patch Set for Linux x86

Used Hardware

- In the VMWare: create one virtual machine with the following specs:
  - 2 GB RAM
  - One ethernet card configured as bridged or host-only in VMware
  - CPU Count: 2
  - Disk1: 40 GB on SCSI 0:0 used to install the OS and software

Installation Steps

1. Hardware ans Software Requirements

Note: A reference to the requirements is Note ID 419646.1.

```
# At least 2 GB of physical memory
grep MemTotal /proc/meminfo

# OS and Kernel version
Oracle Enterprise Manager Grid Control Certification Checker [ID 412431.1]
For OEM 10.2.0.1 in Linux 5.2, only software will be installed (no config)

# swap space: if RAM=2G, swap=4GB
```

```
grep SwapTotal /proc/meminfo

# to display swap and memory in one command:
free

# if you don't have enought swap,
#  you can add swap space by creating a temporary swap file.
#  let's say about 500MB:
dd if=/dev/zero of=tempswap bs=1k count=500000
chmod 600 tempswap
mke2fs tempswap
mkswap tempswap
swapon tempswap


# required packs
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n' binutils \
glibc- \
make- \
gcc- \
libaio- \
glibc-common- \
setarch- \
pdksh- \
openmotif22 \
sysstat- \
libstdc++- \
libstdc++-devel- \
compat-libstdc++- \
compat-db- \
control-center- \


# install missed packages
rpm -Uvh libXp-1.0.0-8.1.el5.i386.rpm
rpm -Uvh openmotif22-2.2.3-18.i386.rpm
rpm -Uvh compat-db-4.2.52-5.1.i386.rpm
rpm -Uvh compat-gcc-34-3.4.6-4.i386.rpm
rpm -Uvh compat-gcc-34-c++-3.4.6-4.i386.rpm

Note:  for pdksh-5.2.14-36.el5.i386.rpm, it isn't used in EL 5.2
```

2. Configure kernel parameters and shell limits

```
vi /etc/sysctl.conf
# mark existing ones
kernel.shmall = 2097152
kernel.shmmax = 536870912
kernel.shmmni = 4096
# semaphores: semmsl, semmns, semopm, semmni
kernel.sem = 250 32000 100 128
# the following setting is removed because default value is higher
# fs.file-max = 65536

# old values (1024 65000) not recommended
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 4194304
net.core.rmem_max = 4194304
```

```
net.core.wmem_default = 262144
net.core.wmem_max = 262144
net.ipv4.tcp_wmem = 262144 262144 262144
net.ipv4.tcp_rmem = 4194304 4194304 4194304

# then run:
/sbin/sysctl -p


vi /etc/security/limits.conf
*     soft    nproc   2047
*     hard    nproc   16384
*     soft    nofile  1024
*     hard    nofile  65536


vi /etc/pam.d/login
session    required     pam_limits.so

# in Linux 5.X run:
ln -s /usr/lib/libgdbm.so.2.0.0 /usr/lib/libdb.so.2

# SELINUX must be disabled
cat /etc/selinux/config | grep SELINUX=
vi /etc/selinux/config
SELINUX=disabled
shutdown -h now -r
```

3. Create the required network configuration:

```
# /etc/hosts file must contain a fully qualified name:
# <IP-address>  <fully-qualified-machine-name>  <machine-name>
vi /etc/hosts
127.0.0.1               localhost.localdomain   localhost
192.168.4.107 srv107.localdomain srv107

ping srv07
```

4. Create and configure the required OS users, groups and directories

```
groupadd -g 501 oinstall
groupadd -g 502 dba
# oracle software owner user
/usr/sbin/useradd -u 200 -g oinstall -G dba oracle
passwd oracle

# oracle parent direcotry
# Oracle homes will be created as subdirectories under this parent directory
mkdir -p /u01/app
mkdir /u01/stage
chown -R oracle:oinstall /u01
chmod 775 /u01/app
chmod 775 /u01/stage

# will be used instead of /tmp by oracle
mkdir /home/oracle/oratemp
chown -R oracle:oinstall /home/oracle/oratemp
```

```
vi /home/oracle/.bash_profile
TMP=/home/oracle/oratemp; export TMP
TMPDIR=$TMP; export TMPDIR
ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
ORACLE_HOME=/u01/app/oracle/product/10.2.0/db10g; export ORACLE_HOME
AGENT_HOME=/u01/app/oracle/product/10.2.0/agent10g; export AGENT_HOME
OMS_HOME=/u01/app/oracle/product/10.2.0/oms10g; export OMS_HOME
ORACLE_HOSTNAME=srv107.localdomain; export ORACLE_HOSTNAME
ORACLE_SID=oemdb; export ORACLE_SID
ORACLE_TERM=xterm; export ORACLE_TERM
PATH=/usr/sbin:$PATH; export PATH
PATH=$ORACLE_HOME/bin:$PATH; export PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export LD_LIBRARY_PATH
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib; export
CLASSPATH


vi /etc/profile
if [ $USER = "oracle" ]; then
        if [ $SHELL = "/bin/ksh" ]; then
              ulimit -p 16384
              ulimit -n 65536
        else
              ulimit -u 16384 -n 65536
        fi
fi
```

5. Install the Software

```
# edit the response file
vi /u01/stage/gc10.0.2.1/Disk1/response/em_with_new_db.rsp

FROM_LOCATION="../rdbms/Disk1/stage/products.xml"
BASEDIR="/u01/app/oracle/product/10.2.0"
INSTALLATION_NAME="OEMGC 10.2.0.5"
s_gdbName="oemdb"
s_mountPoint="/u01/app/oracle/product/10.2.0/oradata"
s_operGroup="dba"
s_adminGroup="dba"
s_securePassword="qwer5"
s_securePasswordConfirm="qwer5"
b_lockedSelected=false
b_passwordsDifferent=false
b_passwordsSame=true
s_reposPwd="qwer5"
s_reposPwdConfirm="qwer5"
UNIX_GROUP_NAME="dba"


# invoke the installer:
# Note: if the folder exists and not empty, add -force switch
cd /u01/stage/gc10.0.2.1/Disk1
./runInstaller -noconfig -silent -responseFile
/u01/stage/gc10.0.2.1/Disk1/response/em_with_new_db.rsp | tee output.txt


# as a root user in a new terminal:
```

```
# (if this is the first Oracle product you just installed on the host):
/u01/app/oraInventory/orainstRoot.sh
# execute (answer n to y/n questions):
/u01/app/oracle/product/10.2.0/db10g/allroot.sh


# stop all the OPMN processes:
/u01/app/oracle/product/10.2.0/oms10g/opmn/bin/opmnctl stopall

# verify the processes were stopped:
ps -ef | grep opmn
```

6. Apply the patches on the repository database home

```
# Apply the interim RDBMS patch# 4329444 to the Oracle home of the database
export ORACLE_HOME=/u01/app/oracle/product/10.2.0/db10g
cd /u01/stage/4329444
export OBJECT_MODE=32_64
export PATH=$PATH:/u01/app/oracle/product/10.2.0/db10g/OPatch
opatch apply


# Apply 10.1.0.5 Patch Set
cd /u01/stage/db10.1.0.5/Disk1
./runInstaller -ignoreSysPrereqs
>select db10g home
>next, next till you finish applying the patch set
```

7. Apply the patch set 10.2.0.5 on OMS and Agent homes

```
# as root: run the following command:
ln -s /usr/lib/libgdbm.so.2.0.0 /usr/lib/libdb.so.2

/* Apply the patch on oms */
# edit the response file:
vi /u01/stage/gc10.0.2.5/3731593/Disk1/response/patchset.rsp
UNIX_GROUP_NAME="dba"
b_softwareonly=true
s_sysPassword="qwer5"
sl_pwdInfo={ "qwer5" }
ORACLE_HOME="/u01/app/oracle/product/10.2.0/oms10g"
oracle.iappserver.st_midtier:szl_InstanceInformation={"ias_password"}
SHOW_SPLASH_SCREEN=false
SHOW_SUMMARY_PAGE=false
SHOW_INSTALL_PROGRESS_PAGE=false
SHOW_REQUIRED_CONFIG_TOOL_PAGE=false
SHOW_CONFIG_TOOL_PAGE=false
SHOW_XML_PREREQ_PAGE=false
HOW_END_OF_INSTALL_MSGS=false
SHOW_ROOTSH_CONFIRMATION=false
SHOW_END_SESSION_PAGE=false

# apply the patch set
cd /u01/stage/gc10.0.2.5/3731593/Disk1
./runInstaller -noconfig -silent -responseFile
/u01/stage/gc10.0.2.5/3731593/Disk1/response/patchset.rsp -force | tee
outputoms.txt
```

```
# open terminal as root and run (no output expected):
/u01/app/oracle/product/10.2.0/oms10g/root.sh


/*  Apply the patch on agent */
# change only the following parameters in the file:
vi /u01/stage/gc10.0.2.5/3731593/Disk1/response/patchset.rsp
ORACLE_HOME="/u01/app/oracle/product/10.2.0/agent10g"

# apply the patch on agent:
cd /u01/stage/gc10.0.2.5/3731593/Disk1
./runInstaller -noconfig -silent -responseFile
/u01/stage/gc10.0.2.5/3731593/Disk1/response/patchset.rsp -force | tee
outputagent.txt

# as root run:
/u01/app/oracle/product/10.2.0/agent10g/root.sh
```

8. Configure the Enterprise Manager Grid Control

```
export PERL5LIB=/u01/app/oracle/product/10.2.0/oms10g/perl/lib/5.6.1

/u01/app/oracle/product/10.2.0/oms10g/perl/bin/perl
/u01/app/oracle/product/10.2.0/oms10g/sysman/install/ConfigureGC.pl
/u01/app/oracle/product/10.2.0

# to verify the configured OEM release:
/u01/app/oracle/product/10.2.0/oms10g/bin/emctl status oms
/u01/app/oracle/product/10.2.0/agent10g/bin/emctl status agent
```

9. Running Enterprise Manager Grid Control 10.2.0.5

```
#find the EM port:
cat /u01/app/oracle/product/10.2.0/oms10g/install/portlist.ini | grep
"Enterprise Manager Central Console Port"


# access the URL:
http://srv107.localdomain:4889/em/


# login as sysman
```

10. Deploy Management Agent on Target Machines and Add them to the Grid Control

- To add a target host to OEM, you need to install Management Agent on the target machine.
- The steps below uses *agentDownload Script* method.

```
/* Install the Agent Software on the Target machine */
# Documenation of the script can be viewed:
http://<OMS_host>:<OMS_port>/agent_download/agent_install_readme.html
http://srv107.localdomain:4889/agent_download/agent_install_readme.html
OR
<OMS_HOME>/sysman/agent_download/agent_install_readme.html

# The agentDownload script is located at
OMS_HOME/sysman/agent_download/<version>/<platform>
```

```
# Download the Management Agent software
# from OTN OR the "Download Agent Software" feature in the Grid Control
# console
Linux_Grid_Control_agent_download_10_2_0_5_0.zip


# In OEM server, copy the Agent downloaded file to:
<OMS_HOME>/sysman/agent_download/10.2.0.5.0


# unzip the file:
unzip Linux_Grid_Control_agent_download_10_2_0_5_0.zip


# Download the agentDownload script to the target host from the Management
Service URL
http://srv107.localdomain:4889/agent_download/10.2.0.2.0/<platform>/agentDownl
oad.<OS>
http://srv107.localdomain:4889/agent_download/10.2.0.5.0/linux/agentDownload.l
inux
OR
scp
oracle@192.168.4.107:/u01/app/oracle/product/10.2.0/oms10g/sysman/agent_downlo
ad/10.2.0.5.0/linux/agentDownload.linux agentDownload.linux

chmod 744 agentDownload.linux


# ensure the wget is there and included in the PATH
ls /usr/local/bin/wget
echo $PATH | grep /usr/local/bin


# ensure the hostnames of the OEM server and target are accessbile from each
# other
ping srv107
ping oradb1


# export the required env variable
export ORACLE_HOSTNAME=oradb1

# RUN THE SCRIPT
# agent10g will be created as a subdirectory
./agentDownload.linux -b /u01/app/oracle/product/10.2.0 -m srv107 -r 4889


# run the script
su root

/u01/app/oracle/product/10.2.0/agent10g/root.sh
exit


# If the agent you are installing is not secure,
# you must execute the following command after the installation is complete:
<Agent_Home>/bin/emctl secure agent <password>
```

```
# status of the agent can be verified by:
cd /u01/app/oracle/product/10.2.0/agent10g/bin
# Note: in the output of the following command, the Agent URL should not be
#  https://localhost:1830/emd/main/. If it is so, OEM won't find the target.
#  You need to re-install the Agent and make sure you follow all the steps
#  above.
./emctl status agent

# also you can check:
https://oradb1:3872/emd/main/

/* Add the Target to OEM */
# login to the OEM and you'll see the target already added:
http://srv107.localdomain:4889/em/
# click on Congiure button to configure the database connection info.
```

Remote Diagnostic Agent (RDA)

Using Remote Diagnostic Agent (RDA)

- Refer to note 314422.1

Connect Oracle to SQL Server

Configuring and Using the Database Control

•

```
To connect to SQL*Server from Oracle using Hetergeneous Services:

1. Install ODBC drivers for the SQL*Server database on the server that
contains the Oracle code tree.

2. Setup the ODBC connection to the SQL Server

3. Test the ODBC conection

4. Ensure the GLOBAL_NAMES parameter is set to FALSE in the Oracle database.
show parameter GLOBAL_NAMES

5. Configure the Hetergeneous services. This consists of creating an
initodbc.ora file in <ORACLE_HOME>\ hs\admin
For example:
 You may create a file named as initclveodbc.ora containing the following:
 #
 # HS init parameters
 #
 HS_FDS_CONNECT_INFO = clve
 HS_FDS_TRACE_LEVEL = off

 #
 # Environment variables required for the non-Oracle system
 #
 #set <envvar>=<value>

7. Modify the Listener.ora file :
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = D:\oracle\product\10.1.0\db_1)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (SID_NAME = CLVEODBC)
      (ORACLE_HOME = D:\oracle\product\10.1.0\db_1)
      (PROGRAM = HSODBC)
    )
    (SID_DESC =
      (SID_NAME = HKCLVEODBC)
      (ORACLE_HOME = D:\oracle\product\10.1.0\db_1)
      (PROGRAM = HSODBC)
    )
  )

8. Modify the TNSNAMES.ORA file to point to the proper code tree.
CLVE.MYDOMAIN.LOCAL =
  (DESCRIPTION =
```

```
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = au-syd-dw01)(PORT = 15220))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = CLVEODBC)
    )
    (HS = OK)
  )

9. Reload the listener
 lsnrctl reload

10. Create a database link on the Oracle installation.
CREATE DATABASE LINK "CLVE.MYDOMAIN.LOCAL"
 CONNECT TO "sqlusername"
 IDENTIFIED BY "sqluserpassword"
 USING 'CLVE.MYDOMAIN.LOCAL';

11. Run a Select statement for the Oracle installation using the database link
select sysdate from dual@CLVE.MYDOMAIN.LOCAL ;
```

Part 12 PL/SQL Samples

PL/SQL Basics

PL/SQL Data Types

```
/* character */
--  max length in 11g: 32767
DECLARE
 v_string VARCHAR2(10);

DECLARE
 v_string VARCHAR2(10 CHAR);
...



/* number */
BINARY_DOUBLE
BINARY_FLOAT
BINARY_INTEGER
NUMBER
PLS_INTEGER     –2147483647 to +2147483647



/* data time */
-- TIMESTAMP
select SYSTIMESTAMP from dual;

-- TIMESTAMP WITH TIME ZONE
SET SERVEROUTPUT ON
DECLARE
 v_datetime TIMESTAMP (3) WITH TIME ZONE := SYSTIMESTAMP;
BEGIN
 DBMS_OUTPUT.PUT_LINE(v_datetime);
END;
/

-- TIMESTAMP WITH LOCAL TIME ZONE
SET SERVEROUTPUT ON
DECLARE
 v_datetime TIMESTAMP (0) WITH LOCAL TIME ZONE := SYSTIMESTAMP;
BEGIN
 DBMS_OUTPUT.PUT_LINE(v_datetime);
END;
/



/* inerval */
INTERVAL YEAR TO MONTH
INTERVAL DAY TO SECOND

DECLARE
v_college_deadline TIMESTAMP;
BEGIN
```

```
 v_college_deadline := TO_TIMESTAMP('06/06/2004', 'DD/MM/YYYY')
  + INTERVAL '12-3' YEAR TO MONTH + INTERVAL '19 9:0:0.0' DAY TO SECOND;
 DBMS_OUTPUT.PUT_LINE('My daughter leaves for college in ' ||
v_college_deadline);
END;
/



/* REF CURSOR */
-- a pointer to a result set
CREATE OR REPLACE PROCEDURE cust_sel
( cv_results IN OUT SYS_REFCURSOR)
IS
BEGIN
 OPEN cv_results FOR
 SELECT customer_id, cust_last_name from customers;
END;
/

VARIABLE x REFCURSOR
EXEC cust_sel(:x)
PRINT x

/* REF */
REF value simply as a pointer to an object instance in an object table or
object view.
```

Controlling Compile-Time Displayed Warnings

```
PLSQL_WARNINGS is a system and session levels.
={ ENABLE | DISABLE | ERROR }:
{ ALL | SEVERE | INFORMATIONAL | PERFORMANCE | { integer | (integer [, integer
] ...) }}

SHOW PARAMETER PLSQL_WARNINGS
-- it can take one value
PLSQL_WARNINGS='ENABLE:PERFORMANCE'

-- it can take multiple values
PLSQL_WARNINGS='ENABLE:PERFORMANCE', 'ENABLE:SEVERE'

-- more examples:
PLSQL_WARNINGS = 'ENABLE:SEVERE', 'DISABLE:INFORMATIONAL';
PLSQL_WARNINGS = 'DISABLE:ALL';
PLSQL_WARNINGS = 'DISABLE:5000', 'ENABLE:5001',
'ERROR:5002';
PLSQL_WARNINGS = 'ENABLE:(5000,5001,5002)',
'DISABLE:(6000,6001)';

-- setting the parameter using DBMS_WARNING
SELECT DBMS_WARNING.GET_WARNING_SETTING_STRING() WARNING_LEVEL FROM dual;

CALL DBMS_WARNING.SET_WARNING_SETTING_STRING('ENABLE:ALL', 'SESSION');
CALL DBMS_WARNING.SET_WARNING_SETTING_STRING('DISABLE:ALL', 'SESSION');
```

Catching Returned Errors

```
..
EXCEPTION
 WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE('Exception: '||sqlerrm);
..
```

Hiding Code

```
wrap iname=input_file.sql oname=output_file.plb
```

Controlling Program Flow

```
/* IF */
IF condition THEN
 action
ELSIF condition THEN
 action
ELSE
 action
END IF;

/* CASE */
CASE expression
 WHEN test1 THEN action;
 WHEN test2 THEN action;
 ...
 ELSE
 ...
END CASE;

CASE v_category
 WHEN 'Oracle Basics'
  THEN v_discount := .15;
 WHEN 'Oracle Server'
  THEN v_discount := .10;
 ELSE v_discount := .5;
END CASE;



/* Searched CASE */
CASE
 WHEN <exp> THEN
  actions;
 WHEN <exp> THEN
  actions;
 ELSE
  actions;
END CASE;



/* Case Expressions */
```

```
appraisal :=
      CASE grade
         WHEN 'A' THEN 'Excellent'
         WHEN 'B' THEN 'Very Good'
         WHEN 'C' THEN 'Good'
         ELSE 'No such grade'
      END;

appraisal :=
     CASE
         WHEN grade = 'A' THEN 'Excellent'
         WHEN grade IN ('B','C') THEN 'Good'
         ELSE 'No such grade'
     END;


/* Loop */
LOOP
 ..
 EXIT [WHEN condition]
END LOOP;

-- loops and lablels
BEGIN
  <<Outer_loop>>
  LOOP
    counter := counter+1;
  EXIT WHEN counter>10;
    <<Inner_loop>>
    LOOP
      ...
      EXIT Outer_loop WHEN total_done = 'YES';
      -- Leave both loops
      EXIT WHEN inner_done = 'YES';
      -- Leave inner loop only
      ...
    END LOOP Inner_loop;
    ...
  END LOOP Outer_loop;
END;
/



/* Numeric FOR Loop */
FOR counter IN [reverse] low_number .. high_number LOOP
 action;
END LOOP;

begin
 for i in reverse 1..10 loop
  dbms_output.put_line(i);
 end loop;
end;
/
```

```
/* WHILE Loop */
WHILE condition LOOP
 ..
END LOOP;


/* Goto and Labels */
BEGIN
 DBMS_OUTPUT.PUT_LINE('BEGINNING OF BLOCK');
GOTO l_Last_Line;
 DBMS_OUTPUT.PUT_LINE('GOTO didn''t work!');
 RETURN;
 <<l_Last_Line>>
 DBMS_OUTPUT.PUT_LINE('Last Line');
END;
/
```

Using Cursors

```
CURSOR cursor_name [parameter_list]
 [RETURN return_type]
 IS query  [FOR UPDATE [OF (column_list)][NOWAIT]];

OPEN cur1;
OPEN cur1(50);

FETCH cursor_name INTO variable_name(s) | PL/SQL_record;

CLOSE cursor_name;

Cursor Attributes:
%BULK_ROWCOUNT  number of rows changed during the operation
%ROWCOUNT number of rows fetched from the cursor at any given time
%FOUND
%ISOPEN
%NOTFOUND
```

```
 -- example 1
SET SERVEROUTPUT ON
DECLARE
 v_first_name AUTHORS.FIRST_NAME%TYPE;
 v_last_name AUTHORS.LAST_NAME%TYPE;
 v_row_count PLS_INTEGER := 0;
 v_book_count PLS_INTEGER := 0;
CURSOR auth_cur IS
 SELECT a.first_name, a.last_name, count(b.title)
 FROM authors a, books b
 WHERE a.id = b.author1
 OR a.id = b.author2
 OR a.id = b.author3
 GROUP BY a.first_name, a.last_name
 HAVING count(b.title) > 0
 ORDER BY a.last_name;
BEGIN
 DBMS_OUTPUT.ENABLE(1000000);
 OPEN auth_cur;
 LOOP
 FETCH auth_cur INTO v_first_name, v_last_name, v_book_count;
  EXIT WHEN auth_cur%NOTFOUND;
  v_row_count := auth_cur%ROWCOUNT;
  DBMS_OUTPUT.PUT_LINE(v_row_count||' rows processed so far');
  DBMS_OUTPUT.PUT_LINE(v_last_name  ||', ' ||v_first_name ||' wrote '
||v_book_count ||' book(s).');
 END LOOP;
 CLOSE auth_cur;
 IF auth_cur%ISOPEN = FALSE THEN
  DBMS_OUTPUT.PUT_LINE('Cursor closed');
 ELSE
  DBMS_OUTPUT.PUT_LINE('The cursor is still open');
 END IF;
 EXCEPTION
```

```
  WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;
/


-- example 2
..
WHILE auth_cur%FOUND LOOP
 DBMS_OUTPUT.PUT_LINE(v_author.last_name);
 FETCH auth_cur INTO v_author;
END LOOP;
..


-- example 3
SET SERVEROUTPUT ON
DECLARE
 CURSOR auth_cur IS
 SELECT * FROM authors;
BEGIN
 FOR v_author IN auth_cur LOOP
  DBMS_OUTPUT.PUT_LINE(v_author.last_name);
 END LOOP;
END;
/



/* Cursor Variables */
SET SERVEROUTPUT ON
DECLARE
 TYPE book_typ IS REF CURSOR RETURN BOOKS%ROWTYPE;
 cv_books book_typ;
 v_books BOOKS%ROWTYPE;
BEGIN
 DBMS_OUTPUT.ENABLE(1000000);
 OPEN cv_books FOR
 SELECT * FROM books WHERE isbn = '78824389';
 FETCH cv_books INTO v_books;
 DBMS_OUTPUT.PUT_LINE(v_books.title||' is '||v_books.price);
 CLOSE cv_books;
END;
/


CREATE OR REPLACE PROCEDURE authors_sel
( cv_results IN OUT SYS_REFCURSOR)
IS
BEGIN
 OPEN cv_results FOR SELECT id, first_name, last_name
 FROM authors;
END;
/
VARIABLE x REFCURSOR
EXEC authors_sel(:x)
PRINT x
```

```
/* Handling Implicit Cursors */
SET SERVEROUTPUT ON
BEGIN
 DBMS_OUTPUT.ENABLE(1000000);
 UPDATE books
  SET price = price * .90
  WHERE isbn = '78824389';
 DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT||' rows updated');
 IF SQL%NOTFOUND THEN
  DBMS_OUTPUT.PUT_LINE('Unable to update isbn 78824389');
 END IF;
 COMMIT;
EXCEPTION
 WHEN OTHERS
 THEN
 DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;
/


/* Cursor Subqueries */
SET SERVEROUTPUT ON
DECLARE
 cv_author SYS_REFCURSOR;
 v_title BOOKS.TITLE%TYPE;
 v_author AUTHORS%ROWTYPE;
 v_counter PLS_INTEGER := 0;
 CURSOR book_cur IS
 SELECT b.title,
 CURSOR (SELECT *
         FROM authors a
         WHERE a.id = b.author1
            OR a.id = b.author2
            OR a.id = b.author3)
 FROM books b WHERE isbn = '78824389';
BEGIN
OPEN book_cur;
LOOP
 FETCH book_cur INTO v_title, cv_author;
 EXIT WHEN book_cur%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE('Title from the main cursor: '||v_title);
 LOOP
 FETCH cv_author INTO v_author;
 EXIT WHEN cv_author%NOTFOUND;
  v_counter := v_counter + 1;
  DBMS_OUTPUT.PUT_LINE('Author'||v_counter||': '
   ||v_author.first_name||' '||v_author.last_name);
 END LOOP;
END LOOP;
CLOSE book_cur;
END;
/
```

```
/* Using Where Current of */
SET SERVEROUTPUT ON
DECLARE
 v_isbn INVENTORY.ISBN%TYPE;
 v_amount INVENTORY.AMOUNT%TYPE;
 CURSOR inventory_cur
 IS
 SELECT isbn, amount
 FROM inventory
 WHERE status = 'IN STOCK'
 AND isbn IN (SELECT isbn
 FROM books
 WHERE price > 40)
 FOR UPDATE OF amount;
BEGIN
 FOR y IN inventory_cur
 LOOP
  FETCH inventory_cur INTO v_isbn, v_amount;
  EXIT WHEN inventory_cur%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE(v_isbn||'Amount IN STOCK before: '||v_amount);
  v_amount := v_amount + 250;
  UPDATE inventory
   SET amount = v_amount
   WHERE CURRENT OF inventory_cur;
  DBMS_OUTPUT.PUT_LINE(v_isbn||'Amount IN STOCK after: '||v_amount);
 END LOOP;
COMMIT;
EXCEPTION
 WHEN OTHERS THEN
 DBMS_OUTPUT.PUT_LINE(SQLERRM);
 ROLLBACK;
END;
/
```

Using Records

```
DECLARE
 TYPE individual_record IS RECORD
 (individual_id INTEGER
  ,first_name VARCHAR2(30 CHAR)
  ,middle_initial individuals.middle_initial%TYPE
  ,last_name VARCHAR2(30 CHAR));

 -- Define a variable of the record type.
 individual INDIVIDUAL_RECORD;
BEGIN
 -- Initialize the field values for the record.
 individual.individual_id := 2;
 individual.first_name := 'John';
 individual.middle_initial := 'P';
 individual.last_name := 'Morgan';

 -- Insert into the table.
 INSERT
 INTO individuals
 VALUES
 (individual.individual_id
  ,individual.first_name
  ,individual.middle_initial
  ,individual.last_name);
  -- Commit the work.
  COMMIT;
END;
/


-- nested records
DECLARE
-- Define a record type.
TYPE individual_record IS RECORD
 (individual_id INTEGER
  ,first_name VARCHAR2(30 CHAR)
  ,middle_initial VARCHAR2(1 CHAR)
  ,last_name VARCHAR2(30 CHAR));

-- Define a record type.
TYPE address_record IS RECORD
(address_id INTEGER
 ,individual_id INTEGER
 ,street_address1 VARCHAR2(30 CHAR)
 ,street_address2 VARCHAR2(30 CHAR)
```

```
 ,street_address3 VARCHAR2(30 CHAR));

-- Define a record type of two user-defined record types.
TYPE individual_address_record IS RECORD
(individual INDIVIDUAL_RECORD
 ,address ADDRESS_RECORD);

-- Define a user-defined compound record type.
individual_address INDIVIDUAL_ADDRESS_RECORD;

BEGIN
-- Initialize the field values for the record.
individual_address.individual.individual_id := 3;
individual_address.individual.first_name := 'Ulysses';
...


-- Record Types can be explicitly defined as Object Type as well


-- Defining and Using Record Types as Return Values
DECLARE
-- Define a record type.
FUNCTION get_row (individual_id_in INTEGER)
RETURN INDIVIDUAL_RECORD IS
 -- Define a cursor to return a row of individuals.
 CURSOR c (individual_id_cursor INTEGER) IS
 SELECT *
 FROM individuals
 WHERE individual_id = individual_id_cursor;
BEGIN
 -- Loop through the cursor for a single row.
 FOR i IN c(individual_id_in) LOOP
  -- Return a %ROWTYPE from the INDIVIDUALS table.
  RETURN i;
 END LOOP;
END get_row;
```

Using Table Functions

```
/* Table Function (not pipelined) */
-- not pipelined: row set will be returned when all rows are fetched
/*  */
CREATE TYPE BookType AS OBJECT (
  isbn  CHAR(10),
  title VARCHAR2(100)
)
/

CREATE TYPE BookTypes AS TABLE OF BookType;
/

CREATE OR REPLACE FUNCTION SomeBooks(p_Category IN books.category%TYPE)
  RETURN BookTypes AS
  v_ResultSet BookTypes := BookTypes();
  CURSOR c_SomeBooks IS
    SELECT isbn, title
      FROM books
      WHERE category = p_Category;

BEGIN
  FOR v_Rec IN c_SomeBooks LOOP
    v_ResultSet.EXTEND;
    v_ResultSet(v_ResultSet.LAST) := BookType(v_Rec.isbn, v_Rec.title);
  END LOOP;
  RETURN v_ResultSet;
END SomeBooks;
/


SELECT SomeBooks('Oracle Basics') FROM dual;

COLUMN title FORMAT a60

SELECT *
  FROM TABLE (SomeBooks('Oracle Basics'));


-- Since SomeBooks will always return the same output given the same
-- input, we can use the DETERMINISTIC keyword.
-- its supposed positive impact is not proved though
CREATE OR REPLACE FUNCTION SomeBooks(p_Category IN books.category%TYPE)
  RETURN BookTypes DETERMINISTIC AS



/* Example of using pipelined table functions in data transformation */
-- with pipelined table functions, each row will be returned as soon
---  as it is created

-- sample destination table
CREATE TABLE yearly_store_sales
 (store_name VARCHAR2(25),
 sales_year NUMBER,
 total_sales NUMBER);

-- (1) create type
CREATE TYPE yearly_store_sales_row AS
 OBJECT(
```

```
 store_name varchar2(25),
 sales_year number,
 total_sales number);

-- (2) create PL/SQL table of the type
CREATE TYPE yearly_store_sales_table
 AS
 TABLE OF yearly_store_sales_row;

-- (3) create package, or function
CREATE OR REPLACE PACKAGE sales_package
AS
 TYPE sales_cursor_type IS REF CURSOR RETURN sales_data%ROWTYPE;

 FUNCTION modify_sales_data (INPUTDATA IN sales_cursor_type)
 RETURN yearly_store_sales_table PIPELINED;

END;
/

CREATE OR REPLACE PACKAGE BODY sales_package
 AS
 FUNCTION modify_sales_data(inputdata IN sales_cursor_type)
 RETURN yearly_store_sales_table PIPELINED
 IS
  inputrec sales_data%ROWTYPE;
  outputrow_2001 yearly_store_sales_row :=
yearly_store_sales_row(NULL,NULL,NULL);
 outputrow_2002 yearly_store_sales_row :=
yearly_store_sales_row(NULL,NULL,NULL);
 BEGIN
 LOOP
  FETCH inputdata INTO inputrec;
  EXIT WHEN inputdata%NOTFOUND;
  IF INPUTREC.SALES_2001 IS NOT NULL THEN
   outputrow_2001.store_name := inputrec.store_name;
   outputrow_2001.sales_year := 2001;
   outputrow_2001.total_sales:= inputrec.sales_2001;
   pipe row (outputrow_2001);
  END IF;
  IF INPUTREC.SALES_2002 IS NOT NULL THEN
   outputrow_2002.store_name := inputrec.store_name;
   outputrow_2002.sales_year := 2002;
   outputrow_2002.total_sales:= inputrec.sales_2002;
   pipe row (outputrow_2002);
  END IF;
 END LOOP;
 RETURN;
 END;
END sales_package;
/

-- (4) usage
SELECT *
 FROM
 TABLE(sales_package.modify_sales_data( CURSOR(select
store_name,sales_2001,sales_2002 FROM sales_data)));
```

Using Collections

- Varrays Fixed: may not be used in tables
- Nested tables Fixed: may be used in tables
- Associative arrays (index-by PL/SQL tables) Dynamic

Multiset Operator	Its Equivalent in SQL
MULTISET EXCEPT	MINUS
MULTISET INTERSECT	INTERSECT
MULTISET UNION	UNION ALL
SET	DISTINCT

Using VARRAYS

- Although VARRAYS can be stored as part of a table, it is best suited for PL/SQL proccessing.

```
TYPE type_name IS {VARRAY | VARYING ARRAY} (size_limit)
OF element_type [ NOT NULL ];
```

```
-- initialized in the declaration section
DECLARE
 TYPE integer_varray IS VARRAY(3) OF INTEGER;
 -- Declare and initialize a varray that allows nulls.
 varray_integer INTEGER_VARRAY := integer_varray(NULL,NULL,NULL);
BEGIN
 -- Assign values to subscripted members of the varray.
 varray_integer(1) := 11;
 varray_integer(2) := 12;
 varray_integer(3) := 13;
 FOR i IN 1..3 LOOP
  dbms_output.put_line('Integer Varray ['||i||'] '||
'['||varray_integer(i)||']');
 END LOOP;
END;
/

-- initialized with zero rows in the declaration
--  then dynamic initialization in the execution
DECLARE
 -- Define a varray of integer with 3 rows.
 TYPE integer_varray IS VARRAY(3) OF INTEGER;
 -- Declare and initialize a null set of rows.
 varray_integer INTEGER_VARRAY := integer_varray();
BEGIN
 -- Loop through the three records.
 FOR i IN 1..3 LOOP
```

```
  -- Initialize row.
  varray_integer.EXTEND;
  -- Assign values to subscripted members of the varray.
  varray_integer(i) := 10 + i;
 END LOOP;
 -- Loop through the records to print the varrays.
 FOR i IN 1..3 LOOP
  -- Print the contents.
  dbms_output.put ('Integer Varray ['||i||'] ');
  dbms_output.put_line('['||varray_integer(i)||']');
 END LOOP;
END;
/


-- Varrays as Object Types
CREATE OR REPLACE TYPE integer_varray
 AS VARRAY(3) OF INTEGER;


-- initializing by Collection API
CREATE OR REPLACE TYPE integer_varray
AS VARRAY(100) OF INTEGER NOT NULL;
/
DECLARE
 -- Declare and initialize a null set of rows.
 varray_integer INTEGER_VARRAY := integer_varray();
 BEGIN
 -- Loop through all records.
 FOR i IN 1..varray_integer.LIMIT LOOP
  -- Initialize row.
  varray_integer.EXTEND;
 END LOOP;

 dbms_output.put ('Integer Varray Initialized ');
 dbms_output.put_line('['||varray_integer.COUNT||']');
END;
/


-- Using Varrays as Column Data Types in Tables
--  it doesn't produce meaningful output
CREATE OR REPLACE TYPE address_varray
AS VARRAY(3) OF VARCHAR2(30 CHAR);
/
CREATE TABLE addresses
(address_id INTEGER NOT NULL
 ..
 ,street_address ADDRESS_VARRAY NOT NULL
```

```
 ..
 );

INSERT INTO addresses
VALUES
(..,address_varray('MyAddress','450 West Paseo Redondo','Suite 200'),..);

-- you can insert only one element of the varray
INSERT
INTO addresses
VALUES (12 ,12 ,address_varray('Office of Senator Kennedy') ,..);

-- it gives though unmeaningful output
SELECT street_address FROM addresses;
STREET_ADDRESS
------------------------------
ADDRESS_VARRAY('MyAddress','450 West Paseo ...

UPDATE addresses
SET street_address =
 address_varray('Your Address'  ,'2400 E. Arizona Biltmore Cir.'
  ,'Suite 1150')
WHERE address_id = 11;

-- to update a portion of a varray column, you must use pl/sql
DECLARE
-- Define a record type for a row of the addresses table.
TYPE address_type IS RECORD .. ;
-- Define a variable of the addresses table record type.
address ADDRESS_TYPE;
-- Define a cursor to return the %ROWTYPE value.
CURSOR get_street_address
 (address_id_in INTEGER) IS
 SELECT *
 FROM addresses
 WHERE address_id = address_id_in;
BEGIN
 -- Open the cursor.
 OPEN get_street_address(11);
 -- Fetch a into the record type variable.
 FETCH get_street_address
 INTO address;
 -- Close the cursor.
 CLOSE get_street_address;
 -- Reset the first element of the varray type variable.
 address.street_address(1) :=
 'Office of Senator John McCain';
 -- Update the varray column value.
 UPDATE addresses
```

```
 SET street_address = address.street_address
 WHERE address_id = 11;
END;
```

Using Nested Tables

- Nested tables may be used in table, record, and object definitions. They may be accessed in SQL and PL/SQL.

```
CREATE OR REPLACE TYPE type_name
AS TABLE OF element_type [ NOT NULL ];
```

```
-- basic example
DECLARE
  -- Define a nested table of variable length strings.
  TYPE card_table IS TABLE OF VARCHAR2(5 CHAR);

  -- Declare and initialize a nested table with three rows.
  cards CARD_TABLE := card_table(NULL,NULL,NULL);
BEGIN
  -- Assign values to subscripted members of the varray.
  cards(1) := 'Ace';
  cards(2) := 'Two';
  cards(3) := 'Three';
END;
/


-- dynamic initialization and assignment in the execution section
DECLARE
  -- Define a nested table of variable length strings.
  TYPE card_suit IS TABLE OF VARCHAR2(5 CHAR);

  -- Declare and initialize a null set of rows.
  cards CARD_SUIT := card_suit();
BEGIN
  -- Loop through the three records.
  FOR i IN 1..3 LOOP
    -- Initialize row.
    cards.EXTEND;
    -- Assign values to subscripted members of the varray.
    IF    i = 1 THEN
      cards(i) := 'Ace';
    ELSIF i = 2 THEN
      cards(i) := 'Two';
    ELSIF i = 3 THEN
      cards(i) := 'Three';
    END IF;
  END LOOP;
END;
/



-- Nested Tables as Column Data Types in Tables
CREATE OR REPLACE TYPE address_table
  AS TABLE OF VARCHAR2(30 CHAR) NOT NULL;
```

```
/

CREATE TABLE addresses
( ..
  ,street_address           ADDRESS_TABLE
  ...)
  NESTED TABLE street_address
  STORE AS nested_street_address;

-- any number of elements can be inserted
INSERT INTO addresses
VALUES
( .. ,address_table('Office of Senator McCain'
  ,'450 West Paseo Redondo','Suite 200'),...);


-- arrange generated output
-- the normal output:
SELECT street_address FROM addresses;
ADDRESS_TABLE('Office of Senator McCain', '450 West Paseo ...
-- to arrange it:
SELECT column_value
FROM THE (SELECT street_address FROM addresses WHERE id = 1);


-- updating a nested-table using pl/sql
CREATE OR REPLACE FUNCTION many_to_one
(street_address_in ADDRESS_TABLE) RETURN VARCHAR2 IS
 -- Define a return variable and initial it.
 retval VARCHAR2(4000) := '';
BEGIN
 -- Loop from the beginning to end of the nested table.
 FOR i IN 1..street_address_in.COUNT LOOP
  -- Append the next value and a line break.
  retval := retval || street_address_in(i) || CHR(10);
 END LOOP;
 RETURN retval;
END many_to_one;
/


-- You can update a portion of a nested table column directly in SQL:
UPDATE THE (SELECT street_address
 FROM addresses
 WHERE address_id = 21)
SET column_value = 'Office of Senator John McCain'
WHERE column_value = 'Office of Senator McCain';
```

Using Associative Arrays

- Associative arrays cannot be used in tables. They are the key to using the FORALL statement or BULK COLLECT clause.
- This collection provides faster and simpler lookups

```
CREATE OR REPLACE TYPE type_name
```

```
AS TABLE OF element_type [ NOT NULL ] INDEX BY [ PLS_INTEGER | BINARY_INTEGER |
VARCHAR2(size) ]
```

```
-- index by integer
DECLARE
  -- Define an associative array of strings.
  TYPE card_table IS TABLE OF VARCHAR2(5 CHAR) INDEX BY BINARY_INTEGER;
  cards CARD_TABLE;
BEGIN
 cards(2) := 'test';
 cards(10) := 'test';
 cards(1) := 'test';
 DBMS_OUTPUT.PUT_LINE(cards.first);
 DBMS_OUTPUT.PUT_LINE(cards.last);
 DBMS_OUTPUT.PUT_LINE(cards.count);
END;
/



-- index by string
..
    -- Loop through all the associative array elements.
    FOR i IN 1..calendar.COUNT LOOP
      -- Check if the first element in the loop.
      IF i = 1 THEN
        -- Assign the first character index to a variable.
        current := calendar.FIRST;
        -- Use the derived index to find the next index.
        element := calendar(current);
      ELSE
        -- Check if next index value exists.
        IF calendar.NEXT(current) IS NOT NULL THEN
          -- Assign the character index to a variable.
          current := calendar.NEXT(current);
          -- Use the derived index to find the next index.
          element := calendar(current);
        ELSE
          -- Exit loop since last index value is read.
          EXIT;
        END IF;
      END IF;
    END LOOP;
..



-- Using Associative Arrays with BULK COLLECT and FORALL
--  BULK COLLECT to retrieve a record set into associative arrays or nested
tables
--  FORALL to send DML statements in batches

-- using FORALL
CREATE TABLE bulk_numbers
(number_id NUMBER
 ,CONSTRAINT number_id_pk  PRIMARY KEY (number_id));

DECLARE
```

```
  TYPE number_table IS TABLE OF bulk_numbers.number_id%TYPE
    INDEX BY BINARY_INTEGER;
  number_list NUMBER_TABLE;
BEGIN

  FOR i IN 1..10000 LOOP
    -- Assign number value.
    number_list(i) := i;
  END LOOP;

  -- Loop through all to do a bulk insert.
  FORALL i IN 1..number_list.COUNT
    INSERT
    INTO     bulk_numbers
    VALUES  (number_list(i));

  COMMIT;
END;
/

-- Using a BULK COLLECT
DECLARE
  TYPE number_table IS TABLE OF bulk_numbers.number_id%TYPE
    INDEX BY BINARY_INTEGER;
  number_list NUMBER_TABLE;
BEGIN
  -- Check if calendar has no elements.
  SELECT   number_id
  BULK COLLECT
  INTO     number_list
  FROM     bulk_numbers
  ORDER BY 1;

  FOR i IN number_list.FIRST..number_list.LAST LOOP
    -- print only the first and last two
    IF i <= 2 OR i >= 9999 THEN
      DBMS_OUTPUT.PUT_LINE('Number ['||number_list(i)||']');
    END IF;
  END LOOP;
END;
/
```

Using Collection API

API Function or Procedure	Notes	Return Type
COUNT		PLS_INTEGER
DELETE(n)		NONE
DELETE(n,m)	n= minimum m=maximum	NONE
EXISTS(n)		TRUE or FALSE
EXTEND	not in Associative Tables	NONE
EXTEND(n)	n = number of elements to add	NONE

EXTEND(n,i)	i= number of replicated elements	NONE
FIRST	the lowest subscript	INTEGER or VARCHAR2
LAST	the highest subscript	INTEGER or VARCHAR2
LIMIT		INTEGER
NEXT(n)	If there is no next element, it will return null.	INTEGER or VARCHAR2
PRIOR(n)	If there is no previous element, it will return null.	INTEGER or VARCHAR2
TRIM	removes the highest subscripted value from a collection	NONE
TRIM(n)	removes the number or elements passed	NONE

Errors returned by Collections:

COLLECTION_IS_NULL	An attempt to use a null collection.
NO_DATA_FOUND	An attempt to use a subscript that has been deleted or is a nonexistent unique string index value in an associative array.
SUBSCRIPT_BEYOND_COUNT	This error applies only to varrays and nested tables.
SUBSCRIPT_OUTSIDE_LIMIT	

```
-- using DELETE method
DECLARE
  -- Define a nested table type of INTEGER.
  TYPE number_table IS TABLE OF INTEGER;
  -- Define a variable of the nested table type.
  number_list NUMBER_TABLE;
  -- Define a local procedure to check and print elements.
  PROCEDURE print_list
    (list_in NUMBER_TABLE) IS
  BEGIN
    -- Loop through the possible index values of the list.
    FOR i IN list_in.FIRST..list_in.LAST LOOP
      -- Check if the subscripted element is there.
      IF list_in.EXISTS(i) THEN
        -- Print the element.
        DBMS_OUTPUT.PUT_LINE('List ['||list_in(i)||']');
      END IF;
    END LOOP;
  END print_list;
BEGIN
  -- Check if a subscript element of one does not exists.
  IF NOT number_list.EXISTS(1) THEN
    -- Construct the collection.
    number_list := number_table(1,2,3,4,5);
  END IF;
  -- Print a title.
  DBMS_OUTPUT.PUT_LINE('Nested table before a deletion');
  DBMS_OUTPUT.PUT_LINE('-----------------------------');
  -- Print the list.
```

```
  print_list(number_list);
  -- Delete an element.
  number_list.DELETE(2,4);
  -- Print a title.
  DBMS_OUTPUT.PUT_LINE(CHR(10)||
                       'Nested table after a deletion');
  DBMS_OUTPUT.PUT_LINE('----------------------------');
  -- Print the list.
  print_list(number_list);
END;
/


-- using EXTEND
SET SERVEROUTPUT ON SIZE 1000000
DECLARE
  -- Define a nested table type of INTEGER.
  TYPE number_table IS TABLE OF INTEGER;
  -- Define a variable of the nested table type.
  number_list NUMBER_TABLE := number_table(1,2);
  -- Define a local procedure to check and print elements.
  PROCEDURE print_list
    (list_in NUMBER_TABLE) IS
  BEGIN
    -- Loop through the possible index values of the list.
    FOR i IN list_in.FIRST..list_in.LAST LOOP
      -- Check if the subscripted element is there.
      IF list_in.EXISTS(i) THEN
        -- Print the element.
        DBMS_OUTPUT.PUT_LINE('List ['||list_in(i)||']');
      END IF;
    END LOOP;
  END print_list;
BEGIN
  -- Print a title.
  DBMS_OUTPUT.PUT_LINE('Nested table before extension');
  DBMS_OUTPUT.PUT_LINE('----------------------------');
  -- Print the list.
  print_list(number_list);
  -- Allocate two null elements.
  number_list.EXTEND(2);
  -- Allocate three elements and copy element two
  number_list.EXTEND(3,2);
  -- Print a title.
  DBMS_OUTPUT.PUT_LINE(CHR(10)||
                       'Nested table after extension');
  DBMS_OUTPUT.PUT_LINE('----------------------------');
  -- Print the list.
  print_list(number_list);
END;
/


-- using LIMIT Method (function)
DECLARE
  -- Define a varray type of INTEGER.
  TYPE number_varray IS VARRAY(5) OF INTEGER;
```

```
  -- Define a variable of the varray type.
  number_list NUMBER_VARRAY := number_varray(1,2,3);
  -- Define a local procedure to check and print elements.
  PROCEDURE print_list
    (list_in NUMBER_VARRAY) IS
  BEGIN
    -- Loop through the possible index values of the list.
    FOR i IN list_in.FIRST..list_in.COUNT LOOP
      -- Print the element.
      DBMS_OUTPUT.PUT_LINE(
        'List Index ['||i||'] '||
        'List Value ['||list_in(i)||']');
    END LOOP;
  END print_list;
BEGIN
  -- Print a title.
  DBMS_OUTPUT.PUT_LINE('Varray after initialization');
  DBMS_OUTPUT.PUT_LINE('--------------------------');
  -- Print the list.
  print_list(number_list);
  -- Extend null element to maximum limit.
  number_list.EXTEND(number_list.LIMIT - number_list.LAST);
  -- Print a title.
  DBMS_OUTPUT.PUT(CHR(10));
  DBMS_OUTPUT.PUT_LINE('Varray after extension');
  DBMS_OUTPUT.PUT_LINE('----------------------');
  -- Print the list.
  print_list(number_list);
END;
/



-- using TRIM Method
DECLARE
  -- Define a varray type of INTEGER.
  TYPE number_varray IS VARRAY(5) OF INTEGER;
  -- Define a variable of the varray type.
  number_list NUMBER_VARRAY := number_varray(1,2,3,4,5);
  -- Define a local procedure to check and print elements.
  PROCEDURE print_list
    (list_in NUMBER_VARRAY) IS
  BEGIN
    -- Loop through the possible index values of the list.
    FOR i IN list_in.FIRST..list_in.COUNT LOOP
      -- Print the element.
      DBMS_OUTPUT.PUT_LINE(
        'List Index ['||i||'] '||
        'List Value ['||list_in(i)||']');
    END LOOP;
  END print_list;
BEGIN
  -- Print a title.
  DBMS_OUTPUT.PUT_LINE('Varray after initialization');
  DBMS_OUTPUT.PUT_LINE('--------------------------');
  -- Print the list.
  print_list(number_list);
```

```
  -- Extend null element to maximum limit.
  number_list.TRIM;
  -- Print a title.
  DBMS_OUTPUT.PUT(CHR(10));
  DBMS_OUTPUT.PUT_LINE('Varray after a single element trim');
  DBMS_OUTPUT.PUT_LINE('---------------------------------');
  -- Print the list.
  print_list(number_list);
  -- Extend null element to maximum limit.
  number_list.TRIM(3);
  -- Print a title.
  DBMS_OUTPUT.PUT(CHR(10));
  DBMS_OUTPUT.PUT_LINE('Varray after a three element trim');
  DBMS_OUTPUT.PUT_LINE('---------------------------------');
  -- Print the list.
  print_list(number_list);
END;
/
```

Handling Errors

Predefined Exceptions

Oracle Error	Equivalent Exception	Description
ORA-0001	DUP_VAL_ON_INDEX	Unique constraint violated.
ORA-0051	TIMEOUT_ON_RESOURCE	Time-out occurred while waiting for resource.
ORA-1001	INVALID_CURSOR	Illegal cursor operation.
ORA-1012	NOT_LOGGED_ON	Not connected to Oracle.
ORA-1017	LOGIN_DENIED	Invalid user name/password.
ORA-1403	NO_DATA_FOUND	No data found.
ORA-1410	SYS_INVALID_ROWID	Conversion to a universal rowid failed.
ORA-1422	TOO_MANY_ROWS	A SELECT.INTO statement matches more than one row.
ORA-1476	ZERO_DIVIDE	Division by zero.
ORA-1722	INVALID_NUMBER	Conversion to a number failed; for example, '1A' is not valid.
ORA-1725	USERENV_COMMITSCN_ERROR1	Incorrect usage of the USERENV('COMMITSCN') function.
ORA-6500	STORAGE_ERROR	Internal PL/SQL error raised if PL/SQL runs out of memory.
ORA-6501	PROGRAM_ERROR	Internal PL/SQL error.
ORA-6502	VALUE_ERROR	Truncation, arithmetic, or conversion error.
ORA-6504	ROWTYPE_MISMATCH	Host cursor variable and PL/SQL cursor variable have incompatible row types.
ORA-6511	CURSOR_ALREADY_OPEN	Attempt to open a cursor that is already open.
ORA-6530	ACCESS_INTO_NULL	Attempt to assign values to the attributes of a NULL object.
ORA-6531	COLLECTION_IS_NULL	Attempt to apply collection methods other than EXISTS to a NULL PL/SQL table or varray.
ORA-6532	SUBSCRIPT_OUTSIDE_LIMIT	Reference to a nested table or varray index outside the declared range (such as –1).
ORA-6533	SUBSCRIPT_BEYOND_COUNT	Reference to a nested table or varray index higher than the number of elements in the collection.
ORA-6548	NO_DATA_NEEDED1	Caller of a pipelined function does not need more rows.
ORA-6592	CASE_NOT_FOUND2	No matching WHEN clause in a CASE statement is found.
ORA-30625	SELF_IS_NULL	Attempt to call a method on a null object instance.

Using User-Defined Exceptions

- It is impossible for an exception handler to be defined for more than one exception simultaneously; i.e.: `WHEN exception1 AND exception2` will raise a compilation error.
- Examining the Error Stack: Use SQLCODE and SQLERRM OR:
- DBMS_UTILITY.FORMAT_ERROR_STACK provides a function FORMAT_ERROR_STACK that returns the same information as SQLERRM, also limited to 2000 bytes.

```
DECLARE
  -- Exception to indicate an error condition
  e_DuplicateAuthors EXCEPTION;
  -- IDs for three authors
  v_Author1 books.author1%TYPE;
  v_Author2 books.author2%TYPE;
  v_Author3 books.author3%TYPE;

BEGIN
  /* Find the IDs for the 3 authors of 'Oracle9i DBA 101' */
  SELECT author1, author2, author3
    INTO v_Author1, v_Author2, v_Author3
    FROM books
    WHERE title = 'Oracle9i DBA 101';

  /* Ensure that there are no duplicates */
  IF (v_Author1 = v_Author2) OR (v_Author1 = v_Author3) OR
     (v_Author2 = v_Author3) THEN
     RAISE e_DuplicateAuthors;
  END IF;
EXCEPTION
  WHEN e_DuplicateAuthors THEN
    INSERT INTO log_table (info)
      VALUES ('Oracle9i DBA 101 has duplicate authors');
  WHEN OTHERS THEN
    INSERT INTO log_table (code, message, info) VALUES
      (NULL, SUBSTR(DBMS_UTILITY.FORMAT_ERROR_STACK, 1, 200),'Oracle error
occurred');
END;
/



/* The EXCEPTION_INIT Pragma */
DECLARE
  e_MissingNull EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_MissingNull, -1400);
BEGIN
  INSERT INTO authors (id) VALUES (NULL);
EXCEPTION
  WHEN e_MissingNull then
    INSERT INTO log_table (info) VALUES ('ORA-1400 occurred');
END;
/
```

Using RAISE_APPLICATION_ERROR

```
RAISE_APPLICATION_ERROR(error_number, error_message, [keep_errors]);
```

- error_number is a value between –20,000 and –20,999
- The error_message must be fewer than 512 characters
- If keep_errors is TRUE, the new error is added to the list of errors already raised (if one exists). If it is FALSE, which is the default, the new error will replace the current list of errors.

```
..
```

```
IF v_AuthorCount = 0 THEN
RAISE_APPLICATION_ERROR(-20001,
'Author1 ' || p_Author1 || ' does not exist');
..
```

Autonomous Transactions

- Autonomous transactions are started by a parent, or main, transaction but operate independently of the parent for transaction control. If a commit or rollback is used in the autonomous or main transaction, or if a failure occurs for any reason, it does not impact the other transaction.

```
CREATE OR REPLACE PROCEDURE logging_ins (
i_username IN VARCHAR2, i_datetime IN TIMESTAMP)
IS
 PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
 INSERT INTO logging (username, datetime)
 VALUES (i_username, i_datetime);
 commit; -- commit is a must here in Autonomous proc othewise ORA-06519
returned
END;
/
```

Some Stored Subprobrams Concepts

Serially Reusable Packages

Serially Reusable Packages	Non–Serially Reusable Packages
Run-time state is kept in shared memory and is freed after every database call.	Run-time state is kept in process memory and lasts for the life of the database session.
The maximum memory used is proportional to the number of concurrent users of the package.	The maximum memory used is proportional to the number of concurrently logged-on users, which is typically much higher.

```
CREATE OR REPLACE PACKAGE TEST_PCK
IS
 -- display number of rows passed
 PROCEDURE DISPLAY_ROWS(P_N IN NUMBER);

END;
/


CREATE OR REPLACE PACKAGE BODY TEST_PCK
IS
 -- this cursor keeps opening in the session life
 CURSOR CR IS SELECT CUSTOMER_ID FROM CUSTOMERS ORDER BY CUSTOMER_ID;

 PROCEDURE DISPLAY_ROWS(P_N IN NUMBER)
 IS
  V_ID NUMBER;
  I NUMBER := 0;
  V_DONE BOOLEAN :=  FALSE;
 BEGIN
  IF NOT CR%ISOPEN THEN
   OPEN CR;
  END IF;
  -- display only passed number of rows
  WHILE NOT V_DONE LOOP
   FETCH CR INTO V_ID;
   IF CR%NOTFOUND THEN
    CLOSE CR;
    V_DONE := TRUE;
   ELSE
    I := I + 1;
    DBMS_OUTPUT.PUT_LINE(V_ID);
    IF I >= P_N THEN
     V_DONE := TRUE;
    END IF;
   END IF;
  END LOOP;
 END DISPLAY_ROWS;
END TEST_PCK;
/


-- if you add PRAGMA SERIALLY_REUSABLE, CR will at reset every call
```

```
CREATE OR REPLACE PACKAGE TEST_PCK
IS
PRAGMA SERIALLY_REUSABLE;
..

CREATE OR REPLACE PACKAGE BODY TEST_PCK
IS
PRAGMA SERIALLY_REUSABLE;
..
```

Stored Subprograms and Roles

- If a stored subprogram refers to an object in another schema, that object should be granted directly to the subprogram owner, not via a role.

```
CONN SA/s

CREATE ROLE R1;

GRANT SELECT ON CUSTOMERS TO R1;

GRANT R1 TO USER1;

CONN USER1/U

CREATE OR REPLACE PACKAGE USER1.TEST_PCK
IS
 PROCEDURE DISPLAY_ROW;
END;
/

-- the following will not compile because CUSTOMERS granted
--  to USER1 via role (not directly)
CREATE OR REPLACE PACKAGE BODY USER1.TEST_PCK
IS
 PROCEDURE DISPLAY_ROW
 IS
  N NUMBER;
 BEGIN
  SELECT CUSTOMER_ID INTO N
  FROM SA.CUSTOMERS
  WHERE CUSTOMER_ID=101;
 END DISPLAY_ROW;
END TEST_PCK;
/
```

Invoker's vs. Definer's Rights

```
CREATE [OR REPLACE] FUNCTION function_name
[parameter_list] RETURN return_type
[AUTHID {CURRENT_USER | DEFINER}] {IS | AS}
```

Pinning an Programunit in the Shared Pool

- To pin an object:

```
DBMS_SHARED_POOL.KEEP(name VARCHAR2, flag CHAR DEFAULT 'P')
flag Determines the type of the object:
 P Package, function, or procedure
 Q Sequence
 R Trigger
 T Object type (Oracle8 and higher)
 JS Java source (Oracle8i and higher)
 JC Java class (Oracle8i and higher)
 JR Java resource (Oracle8i and higher)
 JD Java shared data (Oracle8i and higher)
 C SQL cursor
```

- To echo the contents of the shared pool of objects greater than minisize to the screen:

```
DBMS_SHARED_POOL.SIZES(minsize NUMBER)
```

Using Triggers

- General Syntax of creating any trigger:

```
CREATE [OR REPLACE] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF} triggering_event
[referencing_clause]
[WHEN trigger_condition]
[FOR EACH ROW]
trigger_body
```

Restrictions on Triggers

- A trigger may not issue any transaction control statements—COMMIT, ROLLBACK, SAVEPOINT, or SET TRANSACTION.
- The trigger body cannot declare any LONG or LONG RAW variables.
- Code in a trigger body may reference and use LOB (Large OBject) columns, but it may not modify the values of the columns. This is also true for object columns.

Using DML Triggers

```
CREATE OR REPLACE TRIGGER UpdateCategoryStats
  AFTER INSERT OR DELETE OR UPDATE ON books
DECLARE
  CURSOR c_Statistics IS
    SELECT category,
           COUNT(*) total_books,
           AVG(price) average_price
      FROM books
      GROUP BY category;
BEGIN
  DELETE FROM category_stats;

  FOR v_StatsRecord in c_Statistics LOOP
    INSERT INTO category_stats (category, total_books, average_price)
      VALUES (v_StatsRecord.category, v_StatsRecord.total_books,
              v_StatsRecord.average_price);
  END LOOP;
END UpdateCategoryStats;
/


CREATE OR REPLACE TRIGGER GenerateAuthorID
  BEFORE INSERT OR UPDATE ON authors
  REFERENCING new AS new_author
  FOR EACH ROW
BEGIN
  /* Fill in the ID field of authors with the next value from
     author_sequence. Since ID is a column in authors, :new.ID
     is a valid reference. */
  SELECT author_sequence.NEXTVAL
    INTO :new_author.ID
```

```
    FROM dual;
END GenerateAuthorID;
/


-- using WHEN keyword
CREATE OR REPLACE TRIGGER CheckPrice
 BEFORE INSERT OR UPDATE OF price ON books
 FOR EACH ROW
WHEN (new.price > 49.99)
BEGIN
 ..
END;
/


-- using Trigger Predicates
CREATE OR REPLACE TRIGGER LogInventoryChanges
  BEFORE INSERT OR DELETE OR UPDATE ON inventory
  FOR EACH ROW
DECLARE
  v_ChangeType CHAR(1);
BEGIN
  /* Use 'I' for an INSERT, 'D' for DELETE, and 'U' for UPDATE. */
  IF INSERTING THEN
    v_ChangeType := 'I';
  ELSIF UPDATING THEN
    v_ChangeType := 'U';
  ELSE
    v_ChangeType := 'D';
  END IF;
 ..
END LogInventoryChanges;
/
```

Using Instead-of Triggers

- All instead-of triggers are row level, whether or not the FOR EACH ROW clause is present.

```
CREATE OR REPLACE TRIGGER InsertBooksAuthors
  INSTEAD OF INSERT ON books_authors
DECLARE
  v_Book books%ROWTYPE;
  v_AuthorID authors.id%TYPE;
BEGIN
  -- Figure out the ID of the new author
  BEGIN
    SELECT id
      INTO v_AuthorID
      FROM authors
      WHERE first_name = :new.first_name
        AND last_name = :new.last_name;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      -- No author found, create a new one
      INSERT INTO authors (id, first_name, last_name)
```

```
        VALUES (author_sequence.NEXTVAL, :new.first_name, :new.last_name)
        RETURNING ID INTO v_AuthorID;
  END;
  SELECT *
    INTO v_Book
    FROM books
    WHERE isbn = :new.isbn;
  -- Figure out whether the book already has 1 or 2 authors, and update
  -- accordingly
  IF v_Book.author2 IS NULL THEN
    UPDATE books
      SET author2 = v_AuthorID
      WHERE isbn = :new.isbn;
  ELSE
    UPDATE books
      SET author3 = v_AuthorID
      WHERE isbn = :new.isbn;
  END IF;
END InsertBooksAuthors;
/
```

Using System Triggers

- System trigger fires on two different kinds of events: DDL or database.
- Because system triggers are generally committed anyway, declaring them as autonomous will not have any effect.
- Required privilege on database triggers: ADMINISTER DATABASE TRIGGER

```
CREATE [OR REPLACE] TRIGGER [schema.]trigger_name
 {BEFORE | AFTER}
 {ddl_event_list | database_event_list}
 ON {DATABASE | [schema.]SCHEMA}
 [when_clause]
trigger_body;
```

- Database Events:

Event	When Trigger Fires	Conditions	Restrictions	Transaction	Attribute Functions
STARTUP	When the database is opened.	None allowed	No database operations allowed in the trigger. Return status ignored.	Starts a separate transaction and commits it after firing the triggers.	ora_sysevent ora_login_user ora_instance_num ora_database_name
SHUTDOWN	Just before the server starts the shutdown of an instance. This lets the cartridge shutdown completely. For abnormal instance shutdown, this triiger	None allowed	No database operations allowed in the trigger. Return status ignored.	Starts a separate transaction and commits it after firing the triggers.	ora_sysevent ora_login_user ora_instance_num ora_database_name

Event	When Trigger Fires	Conditions	Restrictions	Transaction	Attribute Functions
	might not fire.				
DB_ROLE_CHANGE	When the database is opened for the first time after a role change.	None allowed	Return status ignored.	Starts a separate transaction and commits it after firing the triggers.	ora_sysevent ora_login_user ora_instance_num ora_database_name
SERVERERROR	When the error eno occurs. If no condition is given, then this trigger fires whenever an error occurs. The trigger does not fire on ORA-1034, ORA-1403, ORA-1422, ORA-1423, and ORA-4030 because they are not true errors or are too serious to continue processing. It also fails to fire on ORA-18 and ORA-20 because a process is not available to connect to the database to record the error.	ERRNO = eno	Depends on the error. Return status ignored.	Starts a separate transaction and commits it after firing the triggers.	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_server_error ora_is_servererror space_error_info

- Schema or Client Events:

Event	When Trigger Fires	Attribute Functions
BEFORE ALTER AFTER ALTER	When a catalog object is altered.	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_type ora_dict_obj_name ora_dict_obj_owner ora_des_encrypted_password (for ALTER USER events) ora_is_alter_column (for ALTER TABLE events) ora_is_drop_column (for ALTER TABLE events)
BEFORE DROP AFTER DROP	When a catalog object is dropped.	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_type ora_dict_obj_name ora_dict_obj_owner
BEFORE ANALYZE AFTER ANALYZE	When an analyze statement is issued	ora_sysevent ora_login_user ora_instance_num ora_database_name

Event	When Trigger Fires	Attribute Functions
		ora_dict_obj_name ora_dict_obj_type ora_dict_obj_owner
BEFORE ASSOCIATE STATISTICS AFTER ASSOCIATE STATISTICS	When an associate statistics statement is issued	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_name ora_dict_obj_type ora_dict_obj_owner ora_dict_obj_name_list ora_dict_obj_owner_list
BEFORE AUDIT AFTER AUDIT BEFORE NOAUDIT AFTER NOAUDIT	When an audit or noaudit statement is issued	ora_sysevent ora_login_user ora_instance_num ora_database_name
BEFORE COMMENT AFTER COMMENT	When an object is commented	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_name ora_dict_obj_type ora_dict_obj_owner
BEFORE CREATE AFTER CREATE	When a catalog object is created.	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_type ora_dict_obj_name ora_dict_obj_owner ora_is_creating_nested_table (for CREATE TABLE events)
BEFORE DDL AFTER DDL	When most SQL DDL statements are issued. Not fired for ALTER DATABASE, CREATE CONTROLFILE, CREATE DATABASE, and DDL issued through the PL/SQL subprogram interface, such as creating an advanced queue.	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_name ora_dict_obj_type ora_dict_obj_owner
BEFORE DISASSOCIATE STATISTICS AFTER DISASSOCIATE STATISTICS	When a disassociate statistics statement is issued	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_name ora_dict_obj_type ora_dict_obj_owner ora_dict_obj_name_list ora_dict_obj_owner_list
BEFORE GRANT AFTER GRANT	When a grant statement is issued	ora_sysevent ora_login_user ora_instance_num ora_database_name

Event	When Trigger Fires	Attribute Functions
		ora_dict_obj_name ora_dict_obj_type ora_dict_obj_owner ora_grantee ora_with_grant_option ora_privileges
BEFORE LOGOFF	At the start of a user logoff	ora_sysevent ora_login_user ora_instance_num ora_database_name
AFTER LOGON	After a successful logon of a user.	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_client_ip_address
BEFORE RENAME AFTER RENAME	When a rename statement is issued.	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_name ora_dict_obj_owner ora_dict_obj_type
BEFORE REVOKE AFTER REVOKE	When a revoke statement is issued	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_name ora_dict_obj_type ora_dict_obj_owner ora_revokee ora_privileges
AFTER SUSPEND	After a SQL statement is suspended because of an out-of-space condition. The trigger must correct the condition so the statement can be resumed.	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_server_error ora_is_servererror space_error_info
BEFORE TRUNCATE AFTER TRUNCATE	When an object is truncated	ora_sysevent ora_login_user ora_instance_num ora_database_name ora_dict_obj_name ora_dict_obj_type ora_dict_obj_owner

```
CREATE OR REPLACE TRIGGER LogCreations
  AFTER CREATE ON SCHEMA
BEGIN
  INSERT INTO ddl_creations (user_id, object_type, object_name,
                             object_owner, creation_date)
    VALUES (USER, ORA_DICT_OBJ_TYPE, ORA_DICT_OBJ_NAME,
```

```
            ORA_DICT_OBJ_OWNER, SYSDATE);
END LogCreations;
```

Handling Mutating Tables in Triggers

- SQL statements in a trigger body may not:
  - Read from or modify any mutating table of the triggering statement. This includes the triggering table itself.
  - Read from or modify the primary-, unique-, or foreign-key columns of a constraining table of the triggering table. They may, however, modify the other columns if desired.
  - These restrictions apply to all row-level triggers. They apply for statement triggers only when the statement trigger would be fired as a result of a DELETE CASCADE operation.
- If an INSERT statement affects only one row, the before- and after-row triggers for that row do not treat the triggering table as mutating. This is the only case where a row-level trigger may read from or modify the triggering table. Statements such as INSERT INTO table SELECT ... always treat the triggering table as mutating, even if the subquery returns only one row.

```
-- the following trigger will return ORA-4091
CREATE OR REPLACE TRIGGER LimitMajors
  BEFORE INSERT OR UPDATE OF major ON students
  FOR EACH ROW
DECLARE
  v_MaxStudents CONSTANT NUMBER := 5;
  v_CurrentStudents NUMBER;
BEGIN
  SELECT COUNT(*)
    INTO v_CurrentStudents
    FROM students
    WHERE major = :new.major;

  -- If there isn't room, raise an error.
  IF v_CurrentStudents + 1 > v_MaxStudents THEN
    RAISE_APPLICATION_ERROR(-20000,
      'Too many students in major ' || :new.major);
  END IF;
END LimitMajors;
/



To workaround: create two triggers: one statement level and the other row
level. The statement trigger store the SELECT result into a PL/SQL table in a
package. The row-level trigger will read that data from the package.

CREATE OR REPLACE PACKAGE StudentData AS
  TYPE t_Majors IS TABLE OF students.major%TYPE INDEX BY BINARY_INTEGER;
  TYPE t_IDs IS TABLE OF students.ID%TYPE INDEX BY BINARY_INTEGER;

  v_StudentMajors t_Majors;
  v_StudentIDs    t_IDs;
  v_NumEntries    BINARY_INTEGER := 0;
END StudentData;
/
```

```
CREATE OR REPLACE TRIGGER RLimitMajors
  BEFORE INSERT OR UPDATE OF major ON students
  FOR EACH ROW
BEGIN
  StudentData.v_NumEntries := StudentData.v_NumEntries + 1;
  StudentData.v_StudentMajors(StudentData.v_NumEntries) := :new.major;
  StudentData.v_StudentIDs(StudentData.v_NumEntries) := :new.id;
END RLimitMajors;
/

CREATE OR REPLACE TRIGGER SLimitMajors
  AFTER INSERT OR UPDATE OF major ON students
DECLARE
  v_MaxStudents     CONSTANT NUMBER := 2;
  v_CurrentStudents NUMBER;
  v_StudentID       students.ID%TYPE;
  v_Major           students.major%TYPE;
BEGIN
  /* Loop through each student inserted or updated, and verify
     that we are still within the limit. */
  FOR v_LoopIndex IN 1..StudentData.v_NumEntries LOOP
    v_StudentID := StudentData.v_StudentIDs(v_LoopIndex);
    v_Major := StudentData.v_StudentMajors(v_LoopIndex);

    -- Determine the current number of students in this major.
    SELECT COUNT(*)
      INTO v_CurrentStudents
      FROM students
      WHERE major = v_Major;

    -- If there isn't room, raise an error.
    IF v_CurrentStudents > v_MaxStudents THEN
      RAISE_APPLICATION_ERROR(-20000,
        'Too many students for major ' || v_Major ||
        ' because of student ' || v_StudentID);
    END IF;
  END LOOP;

  -- Reset the counter so the next execution will use new data.
  StudentData.v_NumEntries := 0;
END SLimitMajors;
```

Dropping and Disabling Triggers

```
DROP TRIGGER <triggername>;

ALTER TRIGGER triggername {DISABLE | ENABLE};

ALTER TABLE authors {DISABLE | ENABLE} ALL TRIGGERS;
```

Using Dynamic SQL

Working with Native Dynamic SQL

```
Using Dynamic SQL
Working with Native Dynamic SQL

-- DDL
statement := 'CREATE SEQUENCE '||sequence_name||CHR(10)
    || ' INCREMENT BY 1'    ||CHR(10)
    || ' START WITH  1'    ||CHR(10)
    || ' CACHE   20'   ||CHR(10)
    || ' ORDER';



-- encapsulates a PL/SQL block SELECT-INTO
PROCEDURE increment_sequence
 ( sequence_name   IN  VARCHAR2,
   sequence_value   IN OUT NUMBER ) IS
 -- Define local native dynamic SQL variables.
 statement     VARCHAR2(2000);
 BEGIN

  -- Build dynamic SQL statement as anonymous block PL/SQL unit.
  statement := 'BEGIN'         ||CHR(10)
    || ' SELECT hr.'||sequence_name||'.nextval'||CHR(10)
    || ' INTO  :retval'     ||CHR(10)
    || ' FROM  DUAL;'      ||CHR(10)
    || 'END;';

  -- Execute dynamic SQL statement.
  EXECUTE IMMEDIATE statement
  USING OUT sequence_value;
END increment_sequence;


-- DML without bind variables (less performance)
PROCEDURE insert_into_table
 ( table_name    IN  VARCHAR2
 , table_column_value1  IN  NUMBER
 , table_column_value2  IN  VARCHAR2
 , table_column_value3  IN  VARCHAR2) IS
 -- Define local variables.
 statement     VARCHAR2(2000);
 BEGIN

  -- Build dynamic SQL statement.
  statement := 'INSERT '
    || 'INTO '||table_name||' '
    || 'VALUES ('
    || ''''||table_column_value1||''','
    || ''''||table_column_value2||''','
    || ''''||table_column_value3||''')';
```

```
  -- Execute the NDS statement.
  EXECUTE IMMEDIATE statement;

  -- Commit the records.
  commit;
END insert_into_table;


-- a DML with ordered bind variables
PROCEDURE inserts_into_table
 ( table_name    IN  VARCHAR2
 , table_column_value1  IN  NUMBER
 , table_column_value2  IN  VARCHAR2
 , table_column_value3  IN  VARCHAR2) IS
 -- Define local variables.
 statement     VARCHAR2(2000);
BEGIN
  -- Build dynamic SQL statement.
  statement := 'INSERT '
    || 'INTO '||table_name||' '
    || 'VALUES (:col_one, :col_two, :col_three)';

  -- Execute the NDS statement.
  EXECUTE IMMEDIATE statement
  USING table_column_value1
  ,  table_column_value2
  ,  table_column_value3;

  -- Commit the records.
  commit;
END inserts_into_table;


-- select single row , single col (DQL)
PROCEDURE single_row_return IS
 -- Define local variables.
 statement     VARCHAR2(2000);
 value_out     VARCHAR2(1);
BEGIN
 -- Build dynamic SQL statement.
 statement := 'SELECT ''A'' FROM DUAL';

 -- Use NDS to query a static string.
 EXECUTE IMMEDIATE statement
  INTO value_out;
END single_row_return;



-- select single row, mutiple col (DQL)
PROCEDURE single_row_return
 ( table_name VARCHAR2
 , column_name1 VARCHAR2
 , column_name2 VARCHAR2
 , column_name3 VARCHAR2 ) IS

 -- Define local variables.
```

```
 statement     VARCHAR2(2000);
 cvalue_out1    VARCHAR2(20);
 cvalue_out2    VARCHAR2(30);
 nvalue_out    NUMBER;
BEGIN
 -- Build dynamic SQL statement.
 statement := 'SELECT '
    || column_name1 ||','
    || column_name2 ||','
    || column_name3 ||' '
    || 'FROM '|| table_name;

EXECUTE IMMEDIATE statement
  INTO nvalue_out, cvalue_out1, cvalue_out2;
END single_row_return;



-- SELECT multiple row of single col (DQL)
-- Create a Varray of a one character string
-- it can also be a local type defined in the procedure
-- it can also be any PL/SQL Collection like PL/SQL table
CREATE OR REPLACE TYPE varchar2_table1 IS
VARRAY(100) OF VARCHAR2(1);
/

PROCEDURE multiple_row_return IS
 -- Define local variables.
 statement     VARCHAR2(2000);
 value_out     VARCHAR2_TABLE1;
BEGIN
 -- using an anonymous block is mandatory, otherwise ORA-03001
 statement := 'BEGIN '
    || 'SELECT ''A'' '
    || 'BULK COLLECT INTO :col_val '
    || 'FROM DUAL;'
    || 'END;';

 -- Use Bulk NDS to query a static string.
  EXECUTE IMMEDIATE statement
  USING OUT value_out;

 -- Use a range loop to read the values.
 FOR i IN 1..value_out.COUNT LOOP
   -- Print output message.
   dbms_output.put_line(value_out(i));
 END LOOP;
END multiple_row_return;



-- multiple row with columns (DQL)
-- NDS can use bulk collections only from within an anonymous-block
CREATE OR REPLACE TYPE card_number_varray IS VARRAY(100) OF NUMBER;
CREATE OR REPLACE TYPE card_name_varray IS VARRAY(100) OF VARCHAR2(2000);
CREATE OR REPLACE TYPE card_suit_varray IS VARRAY(100) OF VARCHAR2(2000);
```

```
PROCEDURE multiple_row_return
 ( table_name VARCHAR2
 , column_name1 VARCHAR2
 , column_name2 VARCHAR2
 , column_name3 VARCHAR2 ) IS
  -- Define local Native Dynamic SQL variables.
  statement     VARCHAR2(2000);
  cvalue_out1    CARD_NAME_VARRAY;
  cvalue_out2    CARD_SUIT_VARRAY;
  nvalue_out    CARD_NUMBER_VARRAY;
 BEGIN
 statement := 'BEGIN '
    || 'SELECT '
    || column_name1 ||','
    || column_name2 ||','
    || column_name3 ||' '
    || 'BULK COLLECT INTO :col1, :col2, :col3 '
    || 'FROM '|| table_name ||';'
    || 'END;';

 -- Execute native dynamic SQL.
 EXECUTE IMMEDIATE statement
  USING OUT nvalue_out, OUT cvalue_out1, OUT cvalue_out2;

  FOR i IN 1..nvalue_out.COUNT LOOP
   dbms_output.put_line('Value from ['||column_name1||'] '|| 'is:
['||nvalue_out(i)||']');
   dbms_output.put_line('Value from ['||column_name1||'] '|| 'is:
['||SUBSTR(cvalue_out1(i),1,20)||']');
   dbms_output.put_line('Value from ['||column_name1||'] '|| 'is:
['||SUBSTR(cvalue_out2(i),1,30)||']');
  END LOOP;
END multiple_row_return;
```

Using DBMS_SQL

- DBMS_SQL still has a major feature that is not delivered in NDS. It does not need to know beforehand the number and types of arguments it will receive and process.

```
To use dbms_sql:
GRANT EXECUTE ON dbms_sys_sql TO SYSTEM WITH GRANT OPTION;
GRANT EXECUTE ON dbms_sql TO SYSTEM WITH GRANT OPTION;


-- Working with DDL and DML Without Bind Variables
-- Procedure to close DBMS_SQL open cursor.
PROCEDURE close_open_cursor ( c IN OUT INTEGER) IS
  BEGIN
    IF dbms_sql.is_open(c) THEN
      dbms_sql.close_cursor(c);
    END IF;
END close_open_cursor;

PROCEDURE create_sequence
 ( sequence_name  IN VARCHAR2) IS
   c               INTEGER := dbms_sql.open_cursor;
   fdbk            INTEGER;
```

```
   statement       VARCHAR2(2000);
BEGIN
      -- Build dynamic SQL statement.
      statement := 'CREATE SEQUENCE '||sequence_name||CHR(10)
                ||  '  INCREMENT BY   1'             ||CHR(10)
                ||  '  START WITH     1'             ||CHR(10)
                ||  '  CACHE          20'            ||CHR(10)
                ||  '  ORDER';

      -- Parse and execute the statement.
      dbms_sql.parse(c,statement,dbms_sql.native);
      fdbk := dbms_sql.execute(c);
      -- Close the open cursor.
      dbms_sql.close_cursor(c);
END create_sequence;


-- encapsulates a PL/SQL block SELECT-INTO.
PROCEDURE increment_sequence
    ( sequence_name           IN     VARCHAR2
    , sequence_value          IN OUT NUMBER  ) IS

    -- Define local DBMS_SQL variables.
    c                         INTEGER := dbms_sql.open_cursor;
    fdbk                      INTEGER;
    statement                 VARCHAR2(2000);

BEGIN
      /*
      || Debugging Tip:
      || =============
      || When you are using a SELECT-INTO-FROM within DBMS_SQL, which is a
      || reserved PLSQL syntax not directly supported by DBMS_SQL.  You
      || need to encapsulate it in a PLSQL wrapper.  When you use a PLSQL
      || wrapper, the semicolons must be used in the statement and the
      || PLSQL block because DBMS_SQL adds a single semicolon to execute
      || the PLSQL block.  If you forget to encapsulate the SQL in a
      || PLSQL wrapper, you will raise the following error message.
      || ------------------------------------------------------------------
      || ORA-01006: bind variable does not exist
      */

      -- Build dynamic SQL statement as anonymous block PL/SQL unit.
      statement := 'BEGIN'||CHR(10)
                ||  ' SELECT PLSQL.'||sequence_name||'.nextval'||CHR(10)
                ||  ' INTO   :retval'||CHR(10)
                ||  ' FROM   DUAL;'||CHR(10)
                ||  'END;';

      -- Parse the statement.
      dbms_sql.parse(c,statement,dbms_sql.native);

      /*
      || Technical Note:
      || ==============
      || The BIND_VARIABLE procedure is returning a NUMBER
      || and does not require parameter four.
```

```
      */

      -- Bind variable retval to an output sequence value.
      dbms_sql.bind_variable(c,'retval',sequence_value);

      -- Execute the dynamic cursor.
      fdbk := dbms_sql.execute(c);

      -- Copy the variable value from the bind variable.
      dbms_sql.variable_value(c,'retval',sequence_value);

      dbms_sql.close_cursor(c);

      dbms_output.put('Sequence <'||sequence_name||'> ');

      dbms_output.put_line('Value <'||sequence_value||'>');
END increment_sequence;



-- a DML with ordered bind variables
  PROCEDURE insert_into_table
    ( table_name              IN     VARCHAR2
    , table_column_value1     IN     NUMBER
    , table_column_value2     IN     VARCHAR2
    , table_column_value3     IN     VARCHAR2) IS

    -- Define local DBMS_SQL variables.
    c                         INTEGER := dbms_sql.open_cursor;
    fdbk                      INTEGER;
    statement                 VARCHAR2(2000);
  BEGIN
  /*
      || Debugging Tip:
      || =============
      || Statement strings are terminated by a line return CHR(10) to
      || ensure that a space is not missing between concatenated segments.
      || Using a BIND variable provides efficiencies in SQL statements
      || because it avoids the reparsing of the statement.  Therefore,
      || they should be used as follows for performance gains:
      ||
      ||   SQL STATEMENTS    PREDICATES
      ||   --------------    ----------
      ||   SELECT            WHERE
      ||   UPDATE            SET
      ||                     WHERE
      ||   DELETE            WHERE
      ||
      || Error Explanations:
      || ------------------
      || 1. An explicit size is always required for a VARCHAR2 variable
      ||    and the overloaded procedure has an output size variable in the
      ||    fourth position that you may need to use.  The output length is
      ||    provided below to demonstrate it.
      || 2. A bad bind variable message typically means the identifier is
      ||    outside of the VARCHAR2 string and treated as a session level
      ||    undefined bind variable.
```

```
      || 3. A "missing SELECT keyword" can occur on an insert statement
      ||    if you put bind variables into the INTO clause for column
      ||    names.
      || 4. If you have quote marks around VARCHAR2 bind variables, you
      ||    may raise the "bind variable does not exist" error.  If you
      ||    need to use that syntax, you can encapsulate the DML in a
      ||    PLSQL wrapper.
      || ------------------------------------------------------------------
      || 1. ORA-06502: PL/SQL: numeric or value error
      || 2. PLS-00049: bad bind variable
      || 3. ORA-00928: missing SELECT keyword
      || 4. ORA-01006: bind variable does not exist
      */

      -- Build dynamic SQL statement.
      statement := 'INSERT '
                || 'INTO '||table_name||' '
                || 'VALUES '
                || '( :table_column_value1'
                || ', :table_column_value2'
                || ', :table_column_value3)';

      -- Parse the statement.
      dbms_sql.parse(c,statement,dbms_sql.native);

      -- Bind each bind variable.
      dbms_sql.bind_variable(c,'table_column_value1',table_column_value1);
      dbms_sql.bind_variable(c,'table_column_value2',table_column_value2);
      dbms_sql.bind_variable(c,'table_column_value3',table_column_value3);

      fdbk := dbms_sql.execute(c);

      dbms_sql.close_cursor(c);

      commit;
      dbms_output.put_line('Value inserted <'||table_column_value1||'>');
      dbms_output.put_line('Value inserted <'||table_column_value2||'>');
      dbms_output.put_line('Value inserted <'||table_column_value3||'>');
END insert_into_table;


-- a DML with ordered bind variables.
PROCEDURE inserts_into_table
    ( table_name              IN     VARCHAR2
    , table_column_values1    IN     DBMS_SQL.NUMBER_TABLE
    , table_column_values2    IN     DBMS_SQL.VARCHAR2_TABLE
    , table_column_values3    IN     DBMS_SQL.VARCHAR2_TABLE) IS

    -- Define local DBMS_SQL variables.
    c                         INTEGER := dbms_sql.open_cursor;
    fdbk                      INTEGER;
    statement                 VARCHAR2(2000);

BEGIN
      -- Build dynamic SQL statement.
      statement := 'INSERT '
                || 'INTO '||table_name||' '
```

```
                || '( card_number '
                || ', card_name '
                || ', card_suit)'
                || 'VALUES '
                || '( :card_number'
                || ', :card_name'
                || ', :card_suit)';

      -- Parse the statement.
      dbms_sql.parse(c,statement,dbms_sql.native);

      -- Bind each bind variable.
      dbms_sql.bind_array(c,'card_number',table_column_values1);
      dbms_sql.bind_array(c,'card_name',table_column_values2);
      dbms_sql.bind_array(c,'card_suit',table_column_values3);

      fdbk := dbms_sql.execute(c);

      dbms_sql.close_cursor(c);

      commit;

      -- Use a for-loop to print values.
      FOR i IN 1..table_column_values1.COUNT LOOP
        dbms_output.put_line(
          'Value inserted <'||table_column_values1(i)||'>');
        dbms_output.put_line(
          'Value inserted <'||table_column_values2(i)||'>');
        dbms_output.put_line(
          'Value inserted <'||table_column_values3(i)||'>');
      END LOOP;
END inserts_into_table;


-- multiple row DQL
PROCEDURE multiple_row_return IS
    -- Define local DBMS_SQL variables.
    c                         INTEGER := dbms_sql.open_cursor;
    fdbk                      INTEGER;
    statement                 VARCHAR2(2000);
    value_out                 VARCHAR2(1);
BEGIN
    statement := 'SELECT ''A'' FROM DUAL';

    dbms_sql.parse(c,statement,dbms_sql.native);

    -- Define the column mapping to the value_out variable.
    dbms_sql.define_column(c,1,value_out,1);

    fdbk := dbms_sql.execute(c);
    LOOP
      -- Exit when no more rows to fetch.
      EXIT WHEN dbms_sql.fetch_rows(c) = 0;
      -- Copy the contents of column #1 to the value_out variable.
      dbms_sql.column_value(c,1,value_out);
      dbms_output.put_line('Value from COLUMN_VALUE <'||value_out||'>');
    END LOOP;
```

```
    -- Close the open cursor.
    dbms_sql.close_cursor(c);
END multiple_row_return;


-- multiple row with columns DQL.
  PROCEDURE multiple_row_return
    ( table_name    VARCHAR2
    , column_name1  VARCHAR2
    , column_name2  VARCHAR2
    , column_name3  VARCHAR2 )IS

    -- Define local DBMS_SQL variables.
    c                         INTEGER := dbms_sql.open_cursor;
    fdbk                      INTEGER;
    statement                 VARCHAR2(2000);
    cvalue_out1               VARCHAR2(2000);
    cvalue_out2               VARCHAR2(2000);
    nvalue_out                NUMBER;

  BEGIN

    -- Build dynamic SQL statement.
    statement := 'SELECT '
              ||  column_name1 ||','
              ||  column_name2 ||','
              ||  column_name3 ||' '
              || 'FROM '|| table_name;

    -- Parse dynamic SQL statement.
    dbms_sql.parse(c,statement,dbms_sql.native);

    /*
    || Debugging Tip:
    || ==============
    || Define the column values and DO NOT forget to assign a size
    || parameter for a string datatype, like VARCHAR2; however, if you
    || forget, the error message is:
    || ------------------------------------------------------------------
    || PLS-00307: too many declarations of 'DEFINE_COLUMN' match this call
    */

    -- Define the column mapping to the value_out variable.
    dbms_sql.define_column(c,1,nvalue_out);
    dbms_sql.define_column(c,2,cvalue_out1,2000);
    dbms_sql.define_column(c,3,cvalue_out2,2000);

    -- Execute dynamic SQL statement.
    fdbk := dbms_sql.execute(c);

    -- Use a loop to read all rows.
    LOOP
      -- Exit when no more rows to fetch.
      EXIT WHEN dbms_sql.fetch_rows(c) = 0;

      -- Copy the contents of column #1 to the value_out variable.
      dbms_sql.column_value(c,1,nvalue_out);
```

```
      dbms_sql.column_value(c,2,cvalue_out1);
      dbms_sql.column_value(c,3,cvalue_out2);

      dbms_output.put_line(
        'Value from ['||column_name1||'] '||
        'is: ['||nvalue_out||']');
      dbms_output.put_line(
        'Value from ['||column_name1||'] '||
        'is: ['||SUBSTR(cvalue_out1,1,5)||']');
      dbms_output.put_line(
        'Value from ['||column_name1||'] '||
        'is: ['||SUBSTR(cvalue_out2,1,8)||']');
    END LOOP;
    dbms_sql.close_cursor(c);
END multiple_row_return;


-- single row DQL.
  /*
  || Demonstrate a single row return using the DEFINE_COLUMN and COLUMN_VALUE
  || program unit, as you would in an explicit cursor.
  */
PROCEDURE single_row_return IS

    -- Define local DBMS_SQL variables.
    c                         INTEGER := dbms_sql.open_cursor;
    fdbk                      INTEGER;
    statement                 VARCHAR2(2000);
    value_out                 VARCHAR2(1);

  BEGIN

    -- Build dynamic SQL statement.
    statement := 'SELECT ''A'' FROM DUAL';

    -- Parse the dynamic SQL statement.
    dbms_sql.parse(c,statement,dbms_sql.native);

    /*
    || Debugging Tip:
    || =============
    || Define the column values and DO NOT forget to assign a size
    || parameter for a string datatype, like VARCHAR2; however, if you
    || forget, the error message is:
    || ------------------------------------------------------------------
    || PLS-00307: too many declarations of 'DEFINE_COLUMN' match this call
    ||
    || This is the message returned because the DEFINE_COLUMN procedure
    || is overloaded and it doesn't know how to implicitly cast without
    || the OUT_VALUE_SIZE argument. Only CHAR, RAW and VARCHAR2 support
    || a fourth argument.
    */

    -- Define the column mapping to the value_out variable.
    dbms_sql.define_column(c,1,value_out,1);

    fdbk := dbms_sql.execute_and_fetch(c);
```

```
    -- Copy the contents of column #1 to the value_out variable.
    dbms_sql.column_value(c,1,value_out);

    dbms_output.put_line(
      'Value from COLUMN_VALUE <'||value_out||'>');

    -- Close the open cursor.
    dbms_sql.close_cursor(c);

END single_row_return;



-- single row DQL.
  PROCEDURE single_row_return
    ( table_name    VARCHAR2
    , column_name1  VARCHAR2
    , column_name2  VARCHAR2
    , column_name3  VARCHAR2 ) IS

    -- Define local DBMS_SQL variables.
    c                         INTEGER := dbms_sql.open_cursor;
    fdbk                      INTEGER;
    statement                 VARCHAR2(2000);
    cvalue_out1               VARCHAR2(20);
    cvalue_out2               VARCHAR2(30);
    nvalue_out                NUMBER;

  BEGIN

    -- Build dynamic SQL statement.
    statement := 'SELECT '
              ||  column_name1 ||','
              ||  column_name2 ||','
              ||  column_name3 ||' '
              || 'FROM '|| table_name;

    -- Parse the dynamic SQL statement.
    dbms_sql.parse(c,statement,dbms_sql.native);

    /*
    || Debugging Tip:
    || =============
    || Define the column values and DO NOT forget to assign a size
    || parameter for a string datatype, like VARCHAR2; however, if you
    || forget, the error message is:
    || ------------------------------------------------------------------
    || PLS-00307: too many declarations of 'DEFINE_COLUMN' match this call
    ||
    || This is the message returned because the DEFINE_COLUMN procedure
    || is overloaded and it doesn't know how to implicitly cast without
    || the OUT_VALUE_SIZE argument. Only CHAR, RAW and VARCHAR2 support
    || a fourth argument.
    */

    -- Define the column mapping to the value_out variable.
```

```
    dbms_sql.define_column(c,1,nvalue_out);
    dbms_sql.define_column(c,2,cvalue_out1,20);
    dbms_sql.define_column(c,3,cvalue_out2,30);

    -- Execute dynamic SQL statement.
    fdbk := dbms_sql.execute_and_fetch(c);

    -- Copy the contents of column #1 to the value_out variable.
    dbms_sql.column_value(c,1,nvalue_out);
    dbms_sql.column_value(c,2,cvalue_out1);
    dbms_sql.column_value(c,3,cvalue_out2);

    -- Print output message.
    dbms_output.put_line('Value from COLUMN_VALUE <'||nvalue_out||'>');
    dbms_output.put_line('Value from COLUMN_VALUE <'||cvalue_out1||'>');
    dbms_output.put_line('Value from COLUMN_VALUE <'||cvalue_out2||'>');

    dbms_sql.close_cursor(c);
END single_row_return;
```

Calling Java from PL/SQL

- Pre-requisits:
  - set CLASSPATH should contain:
    - . current directory
    - Oracle 11g: %ORACLE_HOME%/jdbc/lib/ojdbc5.jar (or 6 but not both)
    - ORACLE_HOME/jlib/orai18n.jar
    - Oracle 10g: %ORACLE_HOME%/jdbc/lib/classes12.zip
    - Java JDK or J2SE 5.n or 6.n
    - In Oracle 11g, the oracle.jdbc.driver.* classes, the ojdbc4.jar file, and the OracleConnectionCacheImpl class are no longer supported or available.
  - Make sure PATH points to JDK home.

```
/* to create a Thick Java client program to Oracle 11g */
-- JDBCExample.java
import java.sql.*;
import oracle.jdbc.pool.OracleDataSource;

public class JDBCExample {
public static void main(String args[]) throws SQLException
 /* Declare the type of Oracle Driver you are using */
 { DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

 /* Create a database connection for the JDBC program */
 Connection conn=
DriverManager.getConnection("jdbc:oracle:thin:@srv01:1521:ora11g","HR","h");
 Statement stmt = conn.createStatement();

 /* Pass a query to SQL and store the results in the result set rs */
 ResultSet rs = stmt.executeQuery("select employee_id, last_name from
employees");

 /* Using the while loop, result set rs is accessed row by row */
 while(rs.next()){
  int number = rs.getInt(1);
  String name= rs.getString(2);
  System.out.println(number+" "+name);
 }

 /* Close the JDBC result set and close the database connection */
 rs.close();
 conn.close();
 }
}


-- load that into the database :
loadjava -r -f -o -user HR/h JDBCExample.class
```

Configuring Oracle Database to Use External Routines

- For further details refer to the documentation or Oracle Database 10g PL/SQL Programming by Ron Hardman, Michael McLaughlin and Scott Urman, Oracle Press.

```
Configure one listener for the database and one for the the extproc agent
#(1) In listener.ora
# remove the IPC protocol from the standard settings
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS =
          (PROTOCOL = TCP)
          (HOST = srv01)
          (PORT = 1521)
        )
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = ora11g)
      (ORACLE_HOME = E:\oracle\OraDB11g)
    )
  )


# another listener. "extproc" lowercase
CALLOUT_LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS =
          (PROTOCOL = IPC)
          (KEY = extproc)
        )
      )
    )
  )


# "PLSExtProc" case sensitive
#       (ENV =
"EXTPROC_DLLS=ONLY:<custom_dll_directory>/<custom_shared_library>,LD_LIBRARY_P
ATH=E:\oracle\OraDB11g\LIB")
SID_LIST_CALLOUT_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = E:\oracle\OraDB11g)
      (PROGRAM = extproc)
      (ENV =
"EXTPROC_DLLS=ONLY:C:\myfiles\dll/C:\myfiles\lib,LD_LIBRARY_PATH=E:\oracle\Ora
```

```
DB11g\LIB")
    )
  )

# if there is an ASM instance
(SID_DESC =
 (GLOBAL_DBNAME = ora11g.srv01)
 (ORACLE_HOME=E:\oracle\OraDB11g\database)
 (SID_NAME = +ASM)
)



#(2) Add the following in to the tnsnames.ora
EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)
      (KEY = extproc))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )


#(3) rebuild the original listener service and create new one
# In Windows
lsnrctl stop
# use NETCA to delete the original service and then re-create it.
# build a new service for the second listener CALLOUT_LISTENER. A new
linstener.ora will
#  be created so paste again the code above in the file.

# In Unix
lsnrctl stop LISTENER
# backup the original linstener.ora and tnsnames files
# copy the new ones
lsnrctl start LISTENER
lsnrctl start CALLOUT_LISTENER
# verify
ps –ef | grep –v grep | grep tnslsnr

#(4) verify the configuration
tnsping EXTPROC_CONNECTION_DATA
If you get a TNS-12541 error when using tnsping, the likelihood is that there
is a mismatch between the ADDRESS parameter values in the listener.ora and
tnsnames.ora files.

# the following command should return the following error:
sqlplus plsql/plsql@EXTPROC_CONNECTION_DATA
ORA-28547: connection to server failed, probable Oracle Net admin error



Working with a C Shared Library
Define a C shared library:
```

```
#include <stdio.h>

/* Declare a writestr function. */
void writestr1(char *path, char *message)
{
  /* Declare a FILE variable. */
  FILE *file_name;

  /* Open the File. */
  file_name = fopen(path,"w");

  /* Write to file the message received. */
  fprintf(file_name,"%s\n",message);

  /* Close the file. */
  fclose(file_name);

}


Unix C Compiler that supports the –G option
cc –G –o writestr1.so writestr1.c
Unix C Compiler that supports the –shared option
cc –shared –o writestr1.so writestr1.c
- OR -
gcc –shared –o writestr1.so writestr1.c


# Defining and Calling the PL/SQL Library Wrapper
CREATE OR REPLACE LIBRARY library_write_string AS
'<oracle_home_directory>/<custom_library>/<file_name>.<file_ext>';
/

CREATE OR REPLACE PROCEDURE write_string
(path VARCHAR2 ,message VARCHAR2) AS EXTERNAL
 LIBRARY library_write_string
 NAME "writestr"
 PARAMETERS (path STRING ,message STRING);
/

-- Available online as part of create_library1.sql
BEGIN
 -- Call the external procedure.
 write_string('/tmp/file.txt','Hello World!');
END;
/
```

Using Large Objects (LOBs)

Creating LOB

- To specify different storage options, use the following syntax in your create table statement:

```
CREATE TABLE table_name ( .. lob_column lob_datatype )
 LOB (lob_column,..) STORE AS [lob_segment_name]
 [(TABLESPACE tablespace
  {ENABLE|DISABLE} STORAGE IN ROW
  STORAGE storage_clause
  CHUNK int
  PCTVERSION int
  CACHE
  CACHE READS [[NO]LOGGING]
  NOCACHE [[NO]LOGGING])
```

- PCTVERSION defaults to 10
- NOCACHE is the default
- ENABLE STORAGE IN ROW, which is the default setting, says that if the LOB value is less than 4K (including control information), store it inline.

```
CREATE TABLE book_samples (
   book_sample_id   NUMBER (10) PRIMARY KEY,
   isbn             CHAR(10 CHAR),
   description      CLOB,
   nls_description  NCLOB,
   book_cover       BLOB,
   chapter_title    VARCHAR2(30 CHAR),
   chapter          BFILE
)
   LOB (book_cover)
      STORE AS blob_seg ( TABLESPACE blob_ts
                  CHUNK 8192
                  PCTVERSION 0
                  NOCACHE
                  NOLOGGING
                  DISABLE STORAGE IN ROW)
   LOB (description, nls_description)
   STORE AS ( TABLESPACE clob_ts
                  CHUNK 8192
                  PCTVERSION 10
                  NOCACHE
                  LOGGING
                  ENABLE STORAGE IN ROW);
```

Using SQL with Internal LOBs

- With SQL, you can insert, update, and delete internal LOBs.
- SQL does not yet support piecewise manipulation of LOBs.

```
SET LONG 64000
```

```
SELECT description FROM book_samples;

-- empty_clob() creates a locator
INSERT INTO book_samples (
   book_sample_id,
   isbn,
   description,
   nls_description,
   book_cover,
   chapter)
 VALUES (
   1,
   '72230665',
   'The ...',
   EMPTY_CLOB(),
   EMPTY_BLOB(),
   BFILENAME('BOOK_SAMPLES_LOC', '72230665.jpg'));

UPDATE book_samples SET description = EMPTY_CLOB() WHERE description IS NOT
NULL;
```

Using LOBs in PL/SQL

- A opened BFILE should be closed after handling it.
- SESSION_MAX_OPEN_FILES controls the number of BFILEs that can be open at any one time.

```
/* APPEND */
CREATE OR REPLACE PROCEDURE LOBAPPEND (
   io_lob_source IN OUT NCLOB,
   io_lob_destination IN OUT NCLOB)
AS
BEGIN
   DBMS_LOB.OPEN(io_lob_source, DBMS_LOB.LOB_READONLY);
   DBMS_LOB.OPEN(io_lob_destination, DBMS_LOB.LOB_READWRITE);

   DBMS_LOB.APPEND(io_lob_destination, io_lob_source);

   -- always close lob after handling them
   DBMS_LOB.CLOSE(io_lob_source);
   DBMS_LOB.CLOSE(io_lob_destination);
END;
/

SET SERVEROUTPUT ON

DECLARE
   v_source_lob_loc NCLOB;
   v_destination_lob_loc NCLOB;
   v_combined_lob NCLOB;
BEGIN
   -- for update is a must here
   SELECT nls_description
   INTO v_source_lob_loc
   FROM book_samples_nls
   FOR UPDATE;
```

```
   SELECT nls_description
   INTO v_destination_lob_loc
   FROM book_samples
   FOR UPDATE;

   LOBAPPEND(v_source_lob_loc, v_destination_lob_loc);

   SELECT nls_description
   INTO v_combined_lob
   from book_samples;

   DBMS_OUTPUT.PUT_LINE(SUBSTR(v_combined_lob, 1, 150));
   DBMS_OUTPUT.PUT_LINE(SUBSTR(v_combined_lob, 151, 300));
END;
/



/* COMPARE */
If 0 is returned, then the LOBs are the same. If 1 is returned, then they are
different.
FUNCTION COMPARE RETURNS NUMBER(38)
Argument Name Type In/Out Default?
_______________ ____________ ___ ____
LOB_1   CLOB IN
LOB_2   CLOB IN
AMOUNT  NUMBER(38) IN DEFAULT
OFFSET_1 NUMBER(38) IN DEFAULT
OFFSET_2 NUMBER(38) IN DEFAULT

CREATE OR REPLACE PROCEDURE CLOB_COMPARE (
   v_lob1 IN OUT CLOB,
   v_lob2 IN OUT CLOB)
AS
   v_compare PLS_INTEGER := 0;
BEGIN
   DBMS_LOB.OPEN(v_lob1, DBMS_LOB.LOB_READONLY);
   DBMS_LOB.OPEN(v_lob2, DBMS_LOB.LOB_READONLY);
   -- buffer size used 32K (the max)
   v_compare := DBMS_LOB.COMPARE(v_lob1,v_lob2, 32768, 1, 1);

   DBMS_OUTPUT.PUT_LINE('The value returned by COMPARE is: '||v_compare);

   DBMS_LOB.CLOSE(v_lob1);
   DBMS_LOB.CLOSE(v_lob2);
END;
/
DECLARE
   v_lob1 CLOB;
   v_lob2 CLOB;
   v_lob3 CLOB;
BEGIN
   SELECT description
   INTO v_lob1
   FROM book_samples
   WHERE book_sample_id = 1;
```

```
   SELECT description
   INTO v_lob2
   FROM book_samples
   WHERE book_sample_id = 2;

   SELECT description
   INTO v_lob3
   FROM book_samples
   WHERE book_sample_id = 3;

   CLOB_COMPARE(v_lob1, v_lob2);

   CLOB_COMPARE(v_lob1, v_lob3);
END;
/



/* CONVERTTO..LOB */
PROCEDURE CONVERTTOBLOB
DEST_LOB        BLOB            IN/OUT
# in CONVERTTOCLOB the following is SRC_BLOB
SRC_CLOB        CLOB            IN
AMOUNT          NUMBER(38)      IN
DEST_OFFSET     NUMBER(38)      IN/OUT
SRC_OFFSET      NUMBER(38)      IN/OUT
BLOB_CSID       NUMBER          IN
LANG_CONTEXT    NUMBER(38)      IN/OUT
WARNING         NUMBER(38)      OUT

CREATE OR REPLACE PROCEDURE CONVERT_ME (
   v_blob_or_clob IN NUMBER, -- if 0 TOBLOB, if 1 TOCLOB
   v_blob IN OUT BLOB,
   v_clob IN OUT CLOB,
   v_amount IN OUT NUMBER,
   v_blob_offset IN OUT NUMBER,
   v_clob_offset IN OUT NUMBER,
   v_lang_context IN OUT NUMBER,
   v_warning OUT NUMBER)
AS
BEGIN

   DBMS_LOB.OPEN(v_blob, DBMS_LOB.LOB_READWRITE);
   DBMS_LOB.OPEN(v_clob, DBMS_LOB.LOB_READWRITE);

   IF v_blob_or_clob = 0
   THEN
   DBMS_LOB.CONVERTTOBLOB(v_blob,
                          v_clob,
                          v_amount,
                          v_blob_offset,
                          v_clob_offset,
                          1,
                          v_lang_context,
                          v_warning);
   ELSE
```

```
   DBMS_LOB.CONVERTTOCLOB(v_clob,
                          v_blob,
                          v_amount,
                          v_clob_offset,
                          v_blob_offset,
                          1,
                          v_lang_context,
                          v_warning);
   END IF;

   DBMS_LOB.CLOSE(v_blob);
   DBMS_LOB.CLOSE(v_clob);

END;
/


DECLARE
   v_clob_or_blob NUMBER;
   v_blob_locator BLOB;
   v_clob_locator CLOB;
   v_blob_offset NUMBER;
   v_clob_offset NUMBER;
   v_lang_context NUMBER := DBMS_LOB.DEFAULT_LANG_CTX;
   v_warning NUMBER;
   v_string_length NUMBER(10);
   v_source_locator BLOB;
   v_destination_locator BLOB;
   v_amount PLS_INTEGER;
   v_string CLOB;
BEGIN

   -- CONVERT CLOB TO BLOB

   SELECT description
   INTO v_clob_locator
   FROM book_samples
   WHERE book_sample_id = 1
   FOR UPDATE;

   SELECT misc
   INTO v_blob_locator
   FROM book_samples
   WHERE book_sample_id = 1
   FOR UPDATE;

   v_string_length := DBMS_LOB.GETLENGTH(v_blob_locator);
   v_amount := DBMS_LOB.GETLENGTH(v_clob_locator);

   DBMS_OUTPUT.PUT_LINE('The initial length of the BLOB is:
'||v_string_length);

        v_clob_or_blob := 0; -- Convert clob to blob
        v_clob_offset := 1;
        v_blob_offset := 1;

        CONVERT_ME(v_clob_or_blob,
```

```
                v_blob_locator,
                v_clob_locator,
                v_amount,
                v_blob_offset,
                v_clob_offset,
                v_lang_context,
                v_warning);

   v_string_length := DBMS_LOB.GETLENGTH(v_blob_locator);

   DBMS_OUTPUT.PUT_LINE('The length of the BLOB post-conversion is:
'||v_string_length);

   -- COPY BLOB FOR ONE ROW TO BLOB IN ANOTHER
   v_source_locator := v_blob_locator;

   SELECT misc
   INTO v_destination_locator
   FROM book_samples
   WHERE book_sample_id = 2
   FOR UPDATE;

   DBMS_LOB.COPY(v_destination_locator, v_source_locator, 32768, 1, 1);

   v_string_length := DBMS_LOB.GETLENGTH(v_destination_locator);

   DBMS_OUTPUT.PUT_LINE('The length of the BLOB post-copy is:
'||v_string_length);

   -- COPY BLOB FOR RECORD 2 BACK TO A CLOB

   SELECT description
   INTO v_clob_locator
   FROM book_samples
   WHERE book_sample_id = 2
   FOR UPDATE;

   SELECT misc
   INTO v_blob_locator
   FROM book_samples
   WHERE book_sample_id = 2
   FOR UPDATE;

   v_string_length := DBMS_LOB.GETLENGTH(v_clob_locator);
   --  v_ammount must equal to BLOB size, otherwise ORA-22993 will return
   v_amount := DBMS_LOB.GETLENGTH(v_blob_locator);

   DBMS_OUTPUT.PUT_LINE('The initial length of the CLOB (record 2) is:
'||v_string_length);

        v_clob_or_blob := 1; -- Convert blob to clob
        -- must be reset to 1 because its value changed by the
        --  previous call of CONVERT_ME
        v_clob_offset := 1;
        v_blob_offset := 1;

        CONVERT_ME(v_clob_or_blob,
                v_blob_locator,
```

```
                v_clob_locator,
                v_amount,
                v_clob_offset,
                v_blob_offset,
                v_lang_context,
                v_warning);

   v_string_length := DBMS_LOB.GETLENGTH(v_clob_locator);

   SELECT description
   INTO v_string
   FROM book_samples
   WHERE book_sample_id = 2;

   DBMS_OUTPUT.PUT_LINE('The length of the CLOB post-conversion is:
'||v_string_length);

   DBMS_OUTPUT.PUT_LINE('The converted CLOB');
   DBMS_OUTPUT.PUT_LINE('==================');
   DBMS_OUTPUT.PUT_LINE(SUBSTR(v_string,1,150));
   DBMS_OUTPUT.PUT_LINE(SUBSTR(v_string,151,300));
END;
/



/* BFILE – FILEEXISTS */
-- This function tests whether a file exists by the name specified in the
insert statement

INSERT INTO book_samples (
   book_sample_id,
   isbn,
   description,
   nls_description,
   misc,
   bfile_description)
 VALUES (
   1,
   '72230665',
   EMPTY_CLOB(),
   EMPTY_CLOB(),
   EMPTY_BLOB(),
   BFILENAME('BOOK_SAMPLES_LOC', 'bfile_example.pdf'));

CREATE OR REPLACE PROCEDURE CHECK_FILE ( v_bfile IN BFILE)
AS
   v_exists PLS_INTEGER := 0;
BEGIN

   v_exists := DBMS_LOB.FILEEXISTS(v_bfile);

   IF v_exists = 0
   THEN
      DBMS_OUTPUT.PUT_LINE ('The file does not exists in the directory
specified');
   ELSE
      DBMS_OUTPUT.PUT_LINE ('The file exists and the directory valid!');
```

```
   END IF;
END;
/

DECLARE
   v_bfile BFILE;
BEGIN
   SELECT bfile_description
   INTO v_bfile
   FROM book_samples
   WHERE book_sample_id = 1;

   CHECK_FILE(v_bfile);
END;
/


/* BFILE – FILEOPEN/OPEN */
-- Oracle recommends that OPEN be used instead of FILEOPEN.
PROCEDURE OPEN
Argument Name Type In/Out Default?
_____________ ___________- ___ ____
FILE_LOC BINARY FILE LOB IN/OUT
OPEN_MODE BINARY_INTEGER IN DEFAULT

OPEN_MODE: DBMS_LOB.LOB_READONLY or DBMS_LOB.LOB_READWRITE



/* BFILE – FILEISOPEN/ISOPEN */
-- ISOPEN should be used in place of FILEISOPEN when possible.
CREATE OR REPLACE PROCEDURE CHECK_STATUS (
   v_bfile IN BFILE)
AS
   v_isopen PLS_INTEGER := 0;
BEGIN
   v_isopen := DBMS_LOB.ISOPEN(v_bfile);
   IF v_isopen = 0
   THEN
      DBMS_OUTPUT.PUT_LINE ('The file is not open.  You must open the');
   ELSE
      DBMS_OUTPUT.PUT_LINE ('The file is open already.');
   END IF;
END;
/
DECLARE
   v_bfile BFILE;
BEGIN
   SELECT bfile_description
   INTO v_bfile
   FROM book_samples
   WHERE book_sample_id = 1;

   CHECK_STATUS(v_bfile);
END;
/
```

```
/* BFILE – FILECLOSE/CLOSE/FILECLOSEALL */
FILECLOSE and CLOSE both close one BFILE at a time, while FILECLOSEALL closes
all open BFILEs. It is recommended by Oracle that CLOSE be used rather than
FILECLOSE for all new development. CLOSE can be used with all LOB types, not
just BFILEs.

CREATE OR REPLACE PROCEDURE CLOSE_ALL_FILES
AS
   v_isopen PLS_INTEGER := 0;
   v_counter PLS_INTEGER := 0;
   v_bfile BFILE;

   CURSOR cur_bfile IS
   SELECT bfile_description
   FROM book_samples;
BEGIN
   DBMS_OUTPUT.PUT_LINE('Open all BFILEs in the table');
   OPEN cur_bfile;
   LOOP
   FETCH cur_bfile INTO v_bfile;
   EXIT WHEN cur_bfile%NOTFOUND;
      BEGIN
         v_counter := v_counter + 1;
         DBMS_LOB.OPEN(v_bfile);
         v_isopen := DBMS_LOB.ISOPEN(v_bfile);
         IF v_isopen = 0
         THEN
            DBMS_OUTPUT.PUT_LINE ('File number '||v_counter||' is closed');
         ELSE
            DBMS_OUTPUT.PUT_LINE ('File number '||v_counter||' is open');
         END IF;
      END;
   END LOOP;

   CLOSE cur_bfile;

   DBMS_LOB.FILECLOSEALL();
   DBMS_OUTPUT.PUT_LINE('         DONE        ');
END;
/



/* LOADFROMFILE/LOADCLOBFROMFILE/LOADBLOBFROMFILE */
load file contents to CLOB and BLOB columns. It is
recommended that LOADCLOBFROMFILE and LOADBLOBFROMFILE be used for
their specific datatypes rather than using the generic overloaded
LOADFROMFILE.

PROCEDURE LOADBLOBFROMFILE
 Argument Name                  Type                    In/Out Default?
 ------------------------------ ----------------------- ------ --------
 DEST_LOB                       BLOB                    IN/OUT
 SRC_BFILE                      BINARY FILE LOB         IN
 AMOUNT                         NUMBER(38)              IN
```

```
 DEST_OFFSET                    NUMBER(38)             IN/OUT
 SRC_OFFSET                     NUMBER(38)             IN/OUT
PROCEDURE LOADCLOBFROMFILE
 Argument Name                  Type                   In/Out Default?
 ------------------------------ ---------------------- ------ --------
 DEST_LOB                       CLOB                   IN/OUT
 SRC_BFILE                      BINARY FILE LOB        IN
 AMOUNT                         NUMBER(38)             IN
 DEST_OFFSET                    NUMBER(38)             IN/OUT
 SRC_OFFSET                     NUMBER(38)             IN/OUT
 BFILE_CSID                     NUMBER                 IN
 LANG_CONTEXT                   NUMBER(38)             IN/OUT
 WARNING                        NUMBER(38)             OUT

set serveroutput on

DECLARE
   v_dest_blob BLOB;
   v_dest_clob CLOB;
   v_source_locator1 BFILE := BFILENAME('BOOK_SAMPLES_LOC',
'bfile_example.pdf');
   v_source_locator2 BFILE := BFILENAME('BOOK_SAMPLES_LOC',
'bfile_example.txt');
   v_source_offset NUMBER := 1;
   v_dest_offset NUMBER := 1;
   v_lang_context NUMBER := DBMS_LOB.DEFAULT_LANG_CTX;
   v_warning PLS_INTEGER;
BEGIN
   -- Empty the description and misc columns
   UPDATE book_samples
    SET description = EMPTY_CLOB(), misc = EMPTY_BLOB()
    WHERE book_sample_id = 1;

   -- Retrieve the locators for the two destination columns
   SELECT       description, misc
   INTO         v_dest_clob, v_dest_blob
   FROM         book_samples
   WHERE book_sample_id = 1
   FOR UPDATE;

   -- Open the BFILEs and destination LOBs
   DBMS_LOB.OPEN(v_source_locator1, DBMS_LOB.LOB_READONLY);
   DBMS_LOB.OPEN(v_source_locator2, DBMS_LOB.LOB_READONLY);
   DBMS_LOB.OPEN(v_dest_blob, DBMS_LOB.LOB_READWRITE);
   DBMS_LOB.OPEN(v_dest_clob, DBMS_LOB.LOB_READWRITE);

   DBMS_OUTPUT.PUT_LINE('Length of the BLOB file is:
'||DBMS_LOB.GETLENGTH(v_source_locator1));
   DBMS_OUTPUT.PUT_LINE('Length of the CLOB file is:
'||DBMS_LOB.GETLENGTH(v_source_locator2));
   DBMS_OUTPUT.PUT_LINE('Size of BLOB pre-load:
'||DBMS_LOB.GETLENGTH(v_dest_blob));
   DBMS_OUTPUT.PUT_LINE('Size of CLOB pre-load:
'||DBMS_LOB.GETLENGTH(v_dest_clob));

   -- Load the destination columns from the source
   DBMS_LOB.LOADBLOBFROMFILE(v_dest_blob, v_source_locator1,
DBMS_LOB.LOBMAXSIZE, v_dest_offset, v_source_offset);
```

```
   DBMS_OUTPUT.PUT_LINE('Size of BLOB post-load: '||(v_dest_offset -1));

   v_dest_offset := 1;
   v_source_offset := 1;

   DBMS_LOB.LOADCLOBFROMFILE(v_dest_clob,
                             v_source_locator2,
                             DBMS_LOB.LOBMAXSIZE,
                             v_dest_offset,
                             v_source_offset,
                             DBMS_LOB.DEFAULT_CSID,
                             v_lang_context,
                             v_warning);

   DBMS_OUTPUT.PUT_LINE('Size of CLOB post-load: '||(v_dest_offset -1));

   -- Close the LOBs that we opened
   DBMS_LOB.CLOSE(v_source_locator1);
   DBMS_LOB.CLOSE(v_source_locator2);
   DBMS_LOB.CLOSE(v_dest_blob);
   DBMS_LOB.CLOSE(v_dest_clob);

EXCEPTION
   WHEN OTHERS
   THEN
      DBMS_OUTPUT.PUT_LINE(SQLERRM);
      DBMS_LOB.CLOSE(v_source_locator1);
      DBMS_LOB.CLOSE(v_source_locator2);
      DBMS_LOB.CLOSE(v_dest_blob);
      DBMS_LOB.CLOSE(v_dest_clob);
END;
/

SET LONG 64000
SELECT description
FROM book_samples
WHERE book_sample_id = 1;
```

Performance Considerations

Using Returning Clause

- RETURNING is a keyword added to the end of the INSERT statement allowing you to work with the LOB immediately, without any additional steps.

```
DECLARE
   v_clob CLOB;
BEGIN
   INSERT INTO book_samples (
   book_sample_id,
   isbn,
   description,
   nls_description,
   book_cover,
   chapter)
   VALUES (
```

```
   1,
   '72230665',
   'The ...',
   EMPTY_CLOB(),
   EMPTY_BLOB(),
   BFILENAME('BOOK_SAMPLES_LOC', '72230665.jpg'))
   RETURNING description INTO v_clob;
   COMMIT;
   DBMS_OUTPUT.PUT_LINE(v_clob);
END;
/
```

Using CONTEXT Index

- You can apply many indexing types. The examples here show how to use CONTEXT index.
- Using the CONTEXT index, we are able to perform the following types of queries:
  - Boolean searches AND, OR, NOT.
  - Exact matches Search for the exact word or phrase inside the text.
  - Inexact matches Search using stemming (a search for mice finds mouse), wildcard, soundex (one word sounds like another).
  - Proximity A word is near another.
  - Ranking A value is provided based on relevance to the keywords used in the query.
  - Theme searches Search on what a document or text is about.
- Refer to Oracle Text documentation for further details.

```
/* Creating CONTEXT Index */
-- 1) create named preferences
-- If you want to index text in another language: eg JAPANESE_LEXER and
JAPANESE_VGRAM_LEXER
-- WORLD_LEXER that can detect the language of the text
BEGIN
 ctx_ddl.create_preference ('lob_lexer', 'basic_lexer');
 ctx_ddl.set_attribute ('lob_lexer', 'index_text', 'true');
 ctx_ddl.set_attribute ('lob_lexer', 'index_themes', 'false');
END;
/

-- 2) create Wordlist
BEGIN
   ctx_ddl.create_preference ('lob_wordlist', 'basic_wordlist');
   ctx_ddl.set_attribute ('lob_wordlist', 'substring_index', 'true');
END;
/

-- 3) Create the index on the column
--  role ctxapp and resource must be granted first
--  STOPLIST refers to noise words. It can also take DEFAULT_STOPLIST
CREATE INDEX lob_indx ON book_samples(description)
 INDEXTYPE IS CTXSYS.CONTEXT
 PARAMETERS ( 'lexer lob_lexer
              wordlist lob_wordlist
              stoplist ctxsys.empty_stoplist' )
/


To examine the tokens:
```

```
set pages 9999
SELECT token_text FROM DR$LOB_INDX$I;


/* Using the Index */
SELECT SCORE(1), book_sample_id
FROM book_samples
WHERE CONTAINS(description, 'website', 1) > 0;
```

Migrating from LONGs to LOBs

```
ALTER TABLE long_to_lob MODIFY text CLOB;
```

PL/SQL Performance Tuning Tips

Use PL/SQL Profiler

- It is DBMS_HPROF in 11g and DBMS_PROFILER in 10g.
- Provides performance statistical information on an executed procedure.

```
/* Required Privs */
GRANT EXECUTE ON dbms_hprof TO sa;
CREATE OR REPLACE DIRECTORY profiler_dir AS '/home/oracle/temp';
GRANT READ, WRITE ON DIRECTORY profiler_dir TO sa;



/* Package Tables */
conn sa/s
@?/rdbms/admin/dbmshptab.sql



/* Using the package */
BEGIN
DBMS_HPROF.start_profiling (
 location => 'PROFILER_DIR',
 filename => 'profiler.txt');
 -- call the procedure to profile
 PROC3;
 DBMS_HPROF.stop_profiling;
END;
/


-- run the ANALYZE function to analyse the
--  raw data and fill the Profiler tables
SET SERVEROUTPUT ON
DECLARE
 l_runid NUMBER;
BEGIN
 l_runid := DBMS_HPROF.analyze (
  location => 'PROFILER_DIR',
  filename => 'profiler.txt',
  run_comment => 'Test run.');
 DBMS_OUTPUT.put_line('l_runid=' || l_runid);
END;
/

l_runid=1


-- check the info:
SELECT runid, run_timestamp, total_elapsed_time, run_comment
FROM dbmshp_runs
WHERE runid=1;
```

```
-- to list the profiling info:
SELECT symbolid, owner, module, type, function
FROM dbmshp_function_info
WHERE runid = 1
ORDER BY symbolid;


-- to list the profiling info in hierarchy:
SELECT RPAD(' ', level*2, ' ') || fi.owner || '.' || fi.module AS name,
    fi.function,
    pci.subtree_elapsed_time,
    pci.function_elapsed_time,
    pci.calls
    FROM dbmshp_parent_child_info pci
    JOIN dbmshp_function_info fi ON pci.runid = fi.runid AND
    pci.childsymid = fi.symbolid
WHERE pci.runid = 1
CONNECT BY PRIOR childsymid = parentsymid
START WITH pci.parentsymid = 3;



/* Using plshprof Utility */
-- profiler.txt generated by the package
plshprof -output plshprof_out profiler.txt
```

Use BULK COLLECT Clause

- More efficient than traditional loop through curoser records

```
/* loading ALL the returned recordset into one PL/SQL tables */
-- if the returned rows are more than 200000, make the rows
--  be returned into batches using LIMIT clause
DECLARE
 -- Define collection type and variables to be used by the
 -- BULK COLLECT clause
 TYPE student_id_type IS TABLE OF student.student_id%TYPE;
 TYPE first_name_type IS TABLE OF student.first_name%TYPE;
 TYPE last_name_type IS TABLE OF student.last_name%TYPE;
 student_id_tab student_id_type;
 first_name_tab first_name_type;
 last_name_tab last_name_type;
BEGIN

 -- Fetch all student data at once via BULK COLLECT clause
 SELECT student_id, first_name, last_name
 BULK COLLECT INTO student_id_tab, first_name_tab, last_name_tab
 FROM student;
 FOR i IN student_id_tab.FIRST..student_id_tab.LAST

 LOOP
  DBMS_OUTPUT.PUT_LINE ('student_id: '||student_id_tab(i));
  DBMS_OUTPUT.PUT_LINE ('first_name: '||first_name_tab(i));
  DBMS_OUTPUT.PUT_LINE ('last_name: '||last_name_tab(i));
 END LOOP;
```

```
END;



/* Using the limit Option */
-- Note: limit option cannot be used directly with SELECT statemnet. Only on
--  cursors
-- batches are good for better performance when
--  the returned rowset is very large ( roughly >500000)
DECLARE
 -- to be used by the LIMIT (batch size)
 v_limit PLS_INTEGER := 50;

 CURSOR student_cur IS
  SELECT student_id, first_name, last_name
  FROM student;

 -- Define collection type and variables to be used by the
 -- BULK COLLECT clause
 TYPE student_id_type IS TABLE OF student.student_id%TYPE;
 TYPE first_name_type IS TABLE OF student.first_name%TYPE;
 TYPE last_name_type IS TABLE OF student.last_name%TYPE;

 student_id_tab student_id_type;
 first_name_tab first_name_type;
 last_name_tab last_name_type;

BEGIN
 OPEN student_cur;
 LOOP
  -- Fetch 50 rows at once
  FETCH student_cur
  BULK COLLECT INTO student_id_tab, first_name_tab,last_name_tab
  LIMIT v_limit;

  -- exit is based on the the number of records in the collection
  EXIT WHEN student_id_tab.COUNT = 0;

  -- it is wrong to put it after the outer loop
  FOR i IN student_id_tab.FIRST..student_id_tab.LAST
  LOOP
   DBMS_OUTPUT.PUT_LINE ('student_id: '||student_id_tab(i));
   DBMS_OUTPUT.PUT_LINE ('first_name: '||first_name_tab(i));
   DBMS_OUTPUT.PUT_LINE ('last_name: '||last_name_tab(i));
  END LOOP;
 END LOOP;
 CLOSE student_cur;
END;


/* Loading into PL/SQL Multiple-Column Table */
-- good when there are so many cols to read
-- Example 1
DECLARE
 CURSOR student_cur IS
  SELECT student_id, first_name, last_name
  FROM student;
```

```
 -- Define record type
 TYPE student_rec IS RECORD
 (student_id student.student_id%TYPE,
 first_name student.first_name%TYPE,
 last_name student.last_name%TYPE);
 -- Define collection type
 TYPE student_type IS TABLE OF student_rec;
 -- Define collection variable
 student_tab student_type;
 -- to be used by the LIMIT clause
 v_limit PLS_INTEGER := 50;

BEGIN
 OPEN student_cur;
 LOOP
  -- Fetch 50 rows at once
  FETCH student_cur BULK COLLECT INTO student_tab LIMIT v_limit;
  EXIT WHEN student_tab.COUNT = 0;

  FOR i IN student_tab.FIRST..student_tab.LAST
  LOOP
   DBMS_OUTPUT.PUT_LINE('student_id: '||student_tab(i).student_id);
   DBMS_OUTPUT.PUT_LINE('first_name: '|| student_tab(i).first_name);
   DBMS_OUTPUT.PUT_LINE('last_name: '|| student_tab(i).last_name);
  END LOOP;
 END LOOP;
 CLOSE student_cur;
END;


-- Example 2:
DECLARE
 V_LIMIT PLS_INTEGER :=50;
 TYPE ORDERS_TYPE IS TABLE OF ORDERS%ROWTYPE;
 ORDERS_TB ORDERS_TYPE;
 CURSOR ORDERS_CRS IS SELECT * FROM ORDERS;
BEGIN
 OPEN ORDERS_CRS;
 LOOP
  -- load 500 at once
  FETCH ORDERS_CRS BULK COLLECT INTO ORDERS_TB LIMIT V_LIMIT;

  -- exit when the collection is empty
  EXIT WHEN ORDERS_TB.COUNT=0;

  -- process the batch
  FOR I IN ORDERS_TB.FIRST .. ORDERS_TB.LAST LOOP
   NULL;
  END LOOP;
 END LOOP;
 CLOSE ORDERS_CRS;
END;
/
```

```
/* Using BULK COLLECT with the DELETE */
DECLARE
 -- Define collection types and variables
 TYPE row_num_type IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
 TYPE row_text_type IS TABLE OF VARCHAR2(10) INDEX BY PLS_INTEGER;
 row_num_tab row_num_type;
 row_text_tab row_text_type;
BEGIN
 DELETE FROM TEST
 RETURNING row_num, row_text
 BULK COLLECT INTO row_num_tab, row_text_tab;

 DBMS_OUTPUT.PUT_LINE ('Deleted '||SQL%ROWCOUNT ||' rows:');

 FOR i IN row_num_tab.FIRST..row_num_tab.LAST
 LOOP
  DBMS_OUTPUT.PUT_LINE ('row_num = '||row_num_tab(i)||
   ' row_text = ' ||row_text_tab(i));
 END LOOP;
 COMMIT;
END;


Output may look like:
Deleted 4 rows:
row_num = 3 row_text = row 3
row_num = 4 row_text = row 4
row_num = 6 row_text = row 6
row_num = 8 row_text = row 8



/* Using BULK COLLECT with FORALL */
-- Example 1: with Insert
DECLARE
 -- Declare collection types
 TYPE string_type IS TABLE OF VARCHAR2(100) INDEX BY PLS_INTEGER;
 TYPE date_type IS TABLE OF DATE INDEX BY PLS_INTEGER;

 -- Declare collection variables to be used by the FORALL statement
 zip_tab string_type;
 city_tab string_type;
 state_tab string_type;
 cr_by_tab string_type;
 cr_date_tab date_type;
 mod_by_tab string_type;
 mod_date_tab date_type;

 v_counter PLS_INTEGER := 0;
 v_total INTEGER := 0;
BEGIN
 -- Populate individual collections
 SELECT *
 BULK COLLECT INTO zip_tab, city_tab, state_tab, cr_by_tab,
 cr_date_tab, mod_by_tab, mod_date_tab
 FROM zipcode
 WHERE state = 'CT';
```

```
 -- Populate MY_ZIPCODE table
 FORALL i in 1..zip_tab.COUNT
 INSERT INTO my_zipcode
 (zip, city, state, created_by, created_date, modified_by,
 modified_date)
 VALUES
 (zip_tab(i), city_tab(i), state_tab(i), cr_by_tab(i),
 cr_date_tab(i), mod_by_tab(i), mod_date_tab(i));
 COMMIT;

 -- Check how many records were added to MY_ZIPCODE table
 SELECT COUNT(*)
 INTO v_total
 FROM my_zipcode
 WHERE state = 'CT';

 DBMS_OUTPUT.PUT_LINE(v_total||' records were added to MY_ZIPCODE table');
END;


-- Example 2: with DELETE
SET SERVEROUTPUT ON
DECLARE
  TYPE order_id_type IS TABLE OF orders.order_id%TYPE;

  oi_in_tb   order_id_type ;
  oi_out_tb  order_id_type ;
BEGIN
  -- Populate collection use in forall.
  SELECT order_id
  BULK COLLECT INTO oi_in_tb
  FROM   orders
  WHERE  order_id between  335332 and 335341;

  FORALL i IN oi_in_tb.first .. oi_in_tb.last
    DELETE FROM orders
    WHERE  order_id = oi_in_tb(i)
    RETURNING order_id BULK COLLECT INTO oi_out_tb;

  DBMS_OUTPUT.put_line('Deleted IDs  : ' || oi_out_tb.count || ' rows');

  ROLLBACK;
END;
```

Set PLSQL_OPTIMIZE_LEVEL and Subprogram Inlining

- The parameter controls level of optimization performed by PL/SQL compiler. It ranges from 0 (no optimization) to 3 (3 in 11g). Thed default (2) is generally acceptable.
- The pragma INLINE compiler directive specifies that a subprogram call is, or is not, to be inlined. It must appear immediately before the subprogram call.

```
SET SERVEROUTPUT ON
DECLARE

 v_num PLS_INTEGER := 1;
```

```
 v_result PLS_INTEGER;

 -- Following are used for elapsed time calculation
 v_start_time NUMBER;
 v_end_time NUMBER;

 -- Define simple function to test PRAGMA INLINE
 FUNCTION test_inline_pragma
  (in_num1 IN PLS_INTEGER, in_num2 IN PLS_INTEGER)
 RETURN PLS_INTEGER
 IS
 BEGIN
  RETURN (in_num1 + in_num2);
 END test_inline_pragma;
 BEGIN

 -- Test function with INLINE PRAGMA enabled
 v_start_time := DBMS_UTILITY.GET_TIME;
 FOR i in 1..10000000 LOOP
  PRAGMA INLINE (test_inline_pragma, 'YES');
  v_result := test_inline_pragma (1, i);
 END LOOP;
 v_end_time := DBMS_UTILITY.GET_TIME;
 DBMS_OUTPUT.PUT_LINE ('Elapsed time when PRAGMA INLINE enabled: '||
 (v_end_time-v_start_time));

 -- Test function with PRAGMA INLINE disabled
 v_start_time := DBMS_UTILITY.GET_TIME;
 FOR i in 1..10000000 LOOP
  PRAGMA INLINE (test_inline_pragma, 'NO');
  v_result := test_inline_pragma (1, i);
 END LOOP;
 v_end_time := DBMS_UTILITY.GET_TIME;
 DBMS_OUTPUT.PUT_LINE ('Elapsed time when INLINE PRAGMA disabled: '||
 (v_end_time-v_start_time));
END;


-- output of the code above:
Elapsed time when PRAGMA INLINE enabled: 46
Elapsed time when INLINE PRAGMA disabled: 147

PL/SQL procedure successfully completed.
```

Using Bind variables in Dynamic SQL

- Using bind variables allows sharable SQL, reduces parse overhead and minimizes latch contention.

```
-- less performance example:
EXECUTE IMMEDIATE 'SELECT COUNT(*) FROM ' || p_mytable || ' WHERE ' ||
p_column || '=' || p_value INTO v_count

-- more efficient
EXECUTE IMMEDIATE 'SELECT COUNT(*) FROM ' || p_mytable || ' WHERE ' ||
p_column || '= :1' INTO v_count USING p_value
```

Use NOCOPY Keyword

- The NOCOPY clause causes a parameter to be passed “by reference” rather than “by value”. Use it, if possible, when the parameter is of a collection datatype.

```
FUNCTION myfunc (
 p_input_table IN OUT NOCOPY number_tab_type,
 p_row  NUMBER, ...
```

Use Associative arrays

- Associative arrays profives faster and simpler lookup than other collections.
- For further examples, see Using Associative arrays.

```
TYPE custs_type IS TABLE OF NUMBER INDEX BY VARCHAR2(1000);
-- the following array should be loaded by some code
g_custs custs_type;

Function get_cust_id ( p_cust_name VARCHAR2)
 RETURN NUMBER
IS
 v_cust_id sh.customers.cust_id%type;
BEGIN
 -- simple and fast lookup (no loops required)
 v_cust_id := g_custs(p_cust_name);
 RETURN (v_cust_id);
END;
```

Use Server Result Cache

- It suits deterministic but expensive functions, such as expensive table lookups on non-volatile tables.
- For details, see Using Server Result Cache.

PL/SQL Miscellaneous Topics

Accessing V$ Views from PL/SQL

```
grant select on V_$SESSION to <username> ;
grant select on V_$PROCESS to <username>;

-- following will NOT work:
GRANT SELECT_CATALOG_ROLE TO <username>;
GRANT SELECT_CATALOG_ROLE TO <username>;

-- connect as <username>
create procedure ..
 select .. from sys.V_$PROCESS p, sys.V_$SESSION s ..
```

Part 13 Appendixes

Program Units and Scripts Used in the Document

Return Parameter Value for Normal User

- Used by normal user who do not have access on v$parameter.

```
CREATE OR REPLACE FUNCTION GET_PAR ( P_PARAMETER VARCHAR2)
RETURN VARCHAR2
IS
  parnam VARCHAR2(256);
  intval BINARY_INTEGER;
  strval VARCHAR2(256);
  partype BINARY_INTEGER;
  V VARCHAR2(1000);
BEGIN
 partype := dbms_utility.get_parameter_value(P_PARAMETER, intval, strval);
  IF partype = 1 THEN
   RETURN  'TYPE:STRING - Value:' || STRVAL;
  ELSE
   RETURN  'TYPE:INTEGER - Value:' || TO_CHAR(intval);
  END IF;
END;
/
```

Applying Random Load on Database Sample 1

- Scripts to apply loads on Oracle DB on a single table. Applicable on Unix and Windows.

Setup

- Database Side

```
/* Create Schema cotaining target objects */
create tablespace hrstbs ;
create user hrs identified by h
default tablespace hrstbs
quota unlimited on hrstbs ;
grant create table, create view, create procedure, create session, create
sequence, plustrace to hrs;
grant execute on DBMS_LOCK to hrs;
grant execute on DBMS_RANDOM to hrs;


/* Create Schema Objects */
conn hrs/h
CREATE TABLE NAMES
( ID NUMBER , NAME VARCHAR2(50), HDATE DATE, SAL NUMBER, REGION VARCHAR2(1));

CREATE SEQUENCE S cache 1000;

/* Create Random Load Package */
-- Generates various different loads on the target database
CREATE OR REPLACE PACKAGE LOAD_GENERATOR
IS
   -- insert batch rows into NAMES table
   PROCEDURE INSERT_NAMES ( P_ROWS IN NUMBER);
   -- high CPU calls
   PROCEDURE SpinCPUs(P_ITERATION IN NUMBER);
   -- high CPU+DB calls
   PROCEDURE SpinCycles (P_ITERATION IN NUMBER);
   -- random query: from NAMES
   PROCEDURE RandomQuery(P_ITERATION IN NUMBER, P_MAX IN NUMBER);
   -- random DML on NAMES
   PROCEDURE RandomDML(P_ITERATION IN NUMBER, P_MAX IN NUMBER);
END load_generator;
/

-- unmark the SLEEP function, if you wish
CREATE OR REPLACE PACKAGE Body LOAD_GENERATOR
IS
   -- generate random text: its lengnth between 4 and the passed value
   FUNCTION G_TEXT(P_SIZE IN NUMBER) RETURN VARCHAR2
    IS
     V VARCHAR2(2000);
    BEGIN
     FOR I IN 1..DBMS_RANDOM.VALUE(4,P_SIZE) LOOP
      V := V || CHR(ROUND(DBMS_RANDOM.VALUE(65,90)));  -- 122
     END LOOP;
     RETURN V;
   END;
```

```
   PROCEDURE INSERT_NAMES ( P_ROWS IN NUMBER)
   IS
    V1 VARCHAR2(15);
    V2 VARCHAR2(15);
   BEGIN
    FOR I IN 1..P_ROWS LOOP
        V1 := G_TEXT(15);
        V2 := G_TEXT(15);
        INSERT INTO NAMES VALUES ( S.NEXTVAL,       -- ID
         V1 || ' ' || V2,                           -- NAME
         TRUNC(SYSDATE-DBMS_RANDOM.VALUE(60,1800)), -- HDATE
         ROUND(DBMS_RANDOM.VALUE(1000,55000)),      -- SAL
         DECODE( TO_CHAR(ROUND(DBMS_RANDOM.VALUE(1,4))),
'1','N','2','W','3','E','4','S') ); -- REGION
      IF MOD(I,100) = 0 THEN
       COMMIT;
      END IF;
    END LOOP;
    COMMIT;
   END INSERT_NAMES;

   PROCEDURE SpinCPUs (P_ITERATION IN NUMBER)
   IS
    N NUMBER;
   BEGIN
    FOR I IN 1.. P_ITERATION LOOP
     -- pure CPU processing (no physical or logical read)
     N := SQRT(ROUND(DBMS_RANDOM.VALUE(1,1000)));
     -- DBMS_LOCK.SLEEP(round(DBMS_RANDOM.VALUE(0.01,0.05),2)); -- in seconds
    END LOOP;
   END SpinCPUs;

   PROCEDURE SpinCycles (P_ITERATION IN NUMBER)
   IS
    N NUMBER;
   BEGIN
    FOR I IN 1.. P_ITERATION LOOP
       FOR X IN 1.. ROUND(DBMS_RANDOM.VALUE(1,10)) LOOP
         N := SQRT(ROUND(DBMS_RANDOM.VALUE(1,1000)));
       END LOOP;
     SELECT COUNT(*) INTO N FROM NAMES;
     -- DBMS_LOCK.SLEEP(ROUND(DBMS_RANDOM.VALUE(0.01,1),2)); -- in seconds
    END LOOP;
   END SpinCycles;

   PROCEDURE RandomQuery(P_ITERATION IN NUMBER, P_MAX IN NUMBER)
   IS
    V_START NUMBER;
    V_END NUMBER;
    N NUMBER;
   BEGIN
    V_END := P_MAX;
    V_START := ROUND(DBMS_RANDOM.VALUE(1,V_END));
    FOR I IN 1.. P_ITERATION LOOP
     SELECT COUNT(ID) INTO N FROM NAMES
      WHERE ID BETWEEN V_START AND V_END;
```

```
     -- DBMS_LOCK.SLEEP(ROUND(DBMS_RANDOM.VALUE(0.01,3),2));
    END LOOP;
   END RandomQuery;

   PROCEDURE RandomDML(P_ITERATION IN NUMBER, P_MAX IN NUMBER)
   IS
    N NUMBER;
    M NUMBER;
    V_NEW_SAL NUMBER;
    V1 VARCHAR2(15);
    V2 VARCHAR2(15);
   BEGIN
    FOR I IN 1.. P_ITERATION LOOP
     N := ROUND(DBMS_RANDOM.VALUE(1,3));
     IF N=1 THEN
        V1 := G_TEXT(15);
        V2 := G_TEXT(15);
        INSERT INTO NAMES VALUES ( S.NEXTVAL,       -- ID
         V1 || ' ' || V2,                           -- NAME
         TRUNC(SYSDATE)-DBMS_RANDOM.VALUE(60,1800), -- HDATE
         ROUND(DBMS_RANDOM.VALUE(1000,55000)),      -- SAL
         DECODE( TO_CHAR(ROUND(DBMS_RANDOM.VALUE(1,4))),
'1','N','2','W','3','E','4','S') ); -- REGION
     ELSIF N=2 THEN
       M := ROUND(DBMS_RANDOM.VALUE(1,P_MAX));
       V_NEW_SAL := ROUND(DBMS_RANDOM.VALUE(1000,55000));
       UPDATE NAMES SET SAL = V_NEW_SAL
        WHERE ID = M;
     ELSIF N=3 THEN
       M := ROUND(DBMS_RANDOM.VALUE(1,P_MAX));
       DELETE NAMES WHERE ID = M;
     END IF;
     -- DBMS_LOCK.SLEEP(ROUND(DBMS_RANDOM.VALUE(0.1,2),2));
     COMMIT;
    END LOOP;
   END RandomDML;
END load_generator;
/


-- load some rows in names table
execute load_generator.insert_names(10000);
```

- OS side: Unix

```
-- in a folder create the following scripts
-- (1) loadcpu1.sh
#!/bin/bash
# apply CPU load on Oracle DB
# parameters: 1 connections, 2 Iterations
users=$1
SRVC="hrserv"
UNPW="hrs/h"
SQLCMD="/home/oracle/scripts/load/loadcpu1.sql"
x=1
y=$users
ITER=$2
```

```
while [ $x -le $y ]
    do
     sqlplus -s $UNPW@$SRVC @$SQLCMD $ITER &
     x=`expr $x + 1`
done

-- (2) loadcpu1.sql
begin
 hrs.LOAD_GENERATOR.SpinCPUs(&1);
end;
/
exit

-- (3) loadcpu2.sh
#!/bin/bash
# apply CPU+DB Calls load on Oracle DB
# parameters: 1 connections, 2 Iterations
users=$1
SRVC="hrserv"
UNPW="hrs/h"
SQLCMD="/home/oracle/scripts/load/loadcpu2.sql"
x=1
y=$users
ITER=$2

while [ $x -le $y ]
    do
     sqlplus -s $UNPW@$SRVC @$SQLCMD $ITER &
     x=`expr $x + 1`
done

-- (4) loadcpu2.sql
begin
 hrs.LOAD_GENERATOR.SpinCycles(&1);
end;
/
exit

-- (5) loadquery.sh
#!/bin/bash
# apply random queries load on Oracle DB
# parameters: 1 connections, 2 Iterations, 3 rows in names
users=$1
SRVC="hrserv"
UNPW="hrs/h"
SQLCMD="/home/oracle/scripts/load/loadquery.sql"
x=1
y=$users
ITER=$2
MAX=$3

while [ $x -le $y ]
    do
     sqlplus -s $UNPW@$SRVC @$SQLCMD $ITER $MAX &
     x=`expr $x + 1`
done
```

```
-- (6) loadquery.sql
begin
 hrs.LOAD_GENERATOR.RANDOMQUERY (&1, &2);
end;
/
exit

-- (7) loaddml.sh
#!/bin/bash
# apply random DML load on Oracle DB
# parameters: 1 connections, 2 Iterations, 3 rows in names
users=$1
SRVC="hrserv"
UNPW="hrs/h"
SQLCMD="/home/oracle/scripts/load/loaddml.sql"
x=1
y=$users
ITER=$2
MAX=$3

while [ $x -le $y ]
    do
     sqlplus -s $UNPW@$SRVC @$SQLCMD $ITER $MAX &
     x=`expr $x + 1`
done

-- (8) loaddml.sql
begin
 hrs.LOAD_GENERATOR.RANDOMDML (&1, &2);
end;
/
exit
```

- OS side: Windows

```
-- in a folder create the following scripts
-- (1) loadcpu1.bat
REM apply CPU load on Oracle DB
REM parameters: 1 connections, 2 Iterations

set users=%1
set SRVC=hrserv
set UNPW=hrs/h
set SQLCMD=C:\TEMP\load\loadcpu1.sql
set x=1
set y=%users%
set ITER=%2

for /L %%i in (1,1,%y%) do (start /d "E:\oracle\OraDB11g" sqlplus -S
%UNPW%@%SRVC% @%SQLCMD% %ITER%)



-- (2) loadcpu1.sql
begin
 hrs.LOAD_GENERATOR.SpinCPUs(&1);
end;
/
```

```
exit


-- (3) loadcpu2.bat
REM apply CPU+DB Calls load on Oracle DB
REM parameters: 1 connections, 2 Iterations

set users=%1
set SRVC=hrserv
set UNPW=hrs/h
set SQLCMD=C:\TEMP\load\loadcpu2.sql
set x=1
set y=%users%
set ITER=%2

for /L %%i in (1,1,%y%) do (start /d "E:\oracle\OraDB11g" sqlplus -S
%UNPW%@%SRVC% @%SQLCMD% %ITER%)


-- (4) loadcpu2.sql
begin
 hrs.LOAD_GENERATOR.SpinCycles(&1);
end;
/
exit



-- (5) loadquery.bat
REM apply random queries load on Oracle DB
REM parameters: 1 connections, 2 Iterations, 3 rows in names

set users=%1
set SRVC=hrserv
set UNPW=hrs/h
set SQLCMD=C:\TEMP\load\loadquery.sql
set x=1
set y=%users%
set ITER=%2
set MAX=%3

for /L %%i in (1,1,%y%) do (start /d "E:\oracle\OraDB11g" sqlplus -S
%UNPW%@%SRVC% @%SQLCMD% %ITER% %MAX%)

-- (6) loadquery.sql
begin
 hrs.LOAD_GENERATOR.RANDOMQUERY (&1, &2);
end;
/
exit


-- (7) loaddml.bat
REM apply random DML load on Oracle DB
REM parameters: 1 connections, 2 Iterations, 3 rows in names
set users=%1
set SRVC=hrserv
```

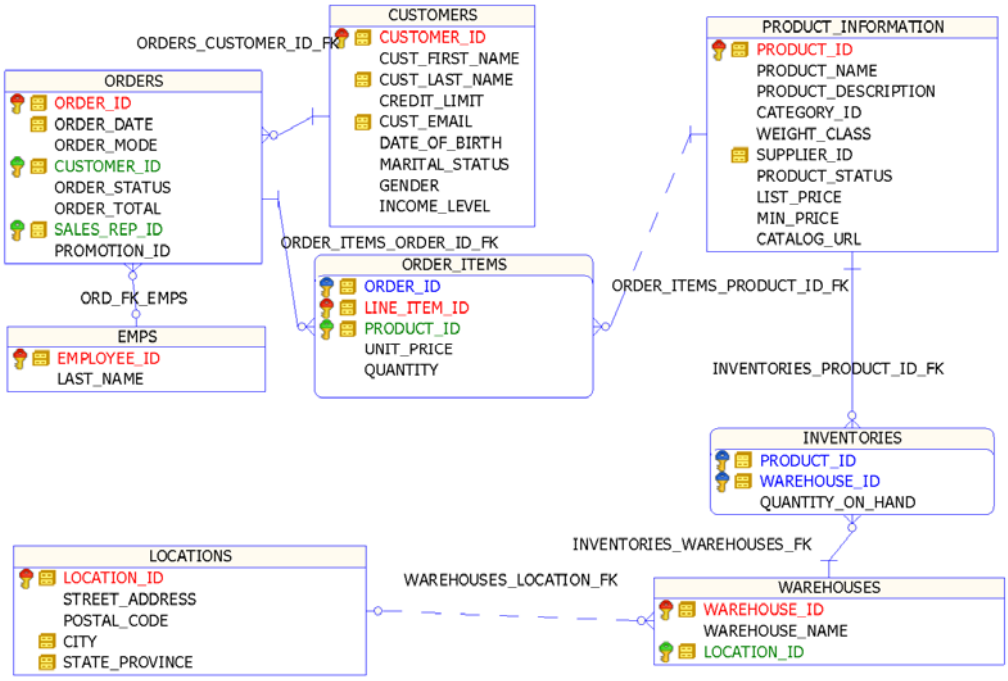
```
set UNPW=hrs/h
set SQLCMD=C:\TEMP\load\loaddml.sql
set x=1
set y=%users%
set ITER=%2
set MAX=%3

for /L %%i in (1,1,%y%) do (start /d "E:\oracle\OraDB11g" sqlplus -S
%UNPW%@%SRVC% @%SQLCMD% %ITER% %MAX%)


-- (8) loaddml.sql
begin
 hrs.LOAD_GENERATOR.RANDOMDML (&1, &2);
end;
/
exit
```

Using the Load Generator Scripts

```
/* in Unix and Windows */
## Using the Load Generator #
# 5 connections 1000 iterations 100000 number of rows in NAMES table
loadcpu1 5 1000
loaddml 5 1000 100000
```

Applying Random Load on Database Sample 2

- Scripts to apply loads on Oracle DB on a schema named as SA. Tables in this schema taken from the default Oracle OE schema. Applicable on Unix and Windows. Following is the ERD of the SA schema:



Setup

- Database Side

```
/* Create sa Schema (if not already there) */
create tablespace satbs
datafile 'C:\ORACLE\ORADATA\ORA11G\satbs1.dbf' size 100m
autoextend on next 12m maxsize 4g
extent management local
segment space management auto ;

-- if ASM is used:
CREATE TABLESPACE satbs DATAFILE '+DATA' size 50m
autoextend on next 12m maxsize 1g;

create user sa identified by s
default tablespace satbs
quota unlimited on satbs ;
grant create table, create view, create procedure, create session, create
sequence to sa;
grant execute on DBMS_LOCK to sa;
grant execute on DBMS_RANDOM to sa;
grant SELECT_CATALOG_ROLE tO sa;
grant alter session to sa;
grant create role to sa;
```

```
grant plustrace to sa;
 -- if role doesn't exist
 @C:\oracle\product\11.1.0\db_1\RDBMS\ADMIN\utlxplan.sql
 @C:\oracle\product\11.1.0\db_1\sqlplus\admin\plustrce.sql
# in unix:
@/u01/app/oracle/product/10.2.0/db_1/rdbms/admin/utlxplan.sql
@/u01/app/oracle/product/10.2.0/db_1/sqlplus/admin/plustrce.sql


/* Create Schema Objects */
conn sa/s
-- execute code in:
@ http://www.ahmedbaraka.com/oracle/scripts/populatesa.sql


-- the package
/* Create Random Load Package */
-- Generates various different loads on the target database
CREATE OR REPLACE PACKAGE sa.LOAD_GENERATOR
IS
   -- global vars
   G_MAX_ORDER_ID NUMBER;
   G_MIN_ORDER_ID NUMBER;

   -- initialize the package
   PROCEDURE INITIALIZE;

   -- insert batch rows into NAMES table
   PROCEDURE INSERT_ORDERS ( P_ROWS IN NUMBER, P_DAYS IN NUMBER DEFAULT 30 );

   -- high CPU calls
   PROCEDURE SpinCPUs(P_ITERATION IN NUMBER);

   -- random query: from ORDERS
   PROCEDURE RandomQuery(P_ITERATION IN NUMBER);

   -- random DML on NAMES
   PROCEDURE RandomDML(P_ITERATION IN NUMBER);
END load_generator;
/


CREATE OR REPLACE PACKAGE Body sa.LOAD_GENERATOR
IS
 PROCEDURE INITIALIZE
 IS
 BEGIN
  SELECT MAX(ORDER_ID) , MIN(ORDER_ID) INTO G_MAX_ORDER_ID, G_MIN_ORDER_ID
FROM ORDERS;
 END INITIALIZE;
   PROCEDURE INSERT_ORDERS ( P_ROWS IN NUMBER, P_DAYS IN NUMBER DEFAULT 30 )
   IS
    V_ORDER_ID NUMBER;
    V_ORDER_DATE DATE;
    V_ORDER_MODE VARCHAR2(8);
    V_CUSTOMER_ID NUMBER;
```

```
    V_ORDER_STATUS NUMBER;
    V_SALES_REP_ID NUMBER;
    V_PRODUCT_ID NUMBER;
    V_PROD_PRICE NUMBER;
    V_CHANGE_PRICE NUMBER;
    V_QUANTITY NUMBER;
    V_ORDER_TOTAL NUMBER;
    V_ITEMS_COUNT NUMBER;
    N NUMBER;
    M NUMBER;
    -- 97 customers
    TYPE CUST_ID_TYPE IS VARRAY(100) OF INTEGER;
    V_CUST_IDS CUST_ID_TYPE := CUST_ID_TYPE
(341,342,343,344,345,346,347,348,349,350,351,352,360,361,363,378,380,447,448,4
49,450,451,452,453,454,458,463,466,467,468,470,473,474,475,476,477,478,479,480
,481,482,483,487,488,492,496,605,606,607,609,615,621,627,712,713,715,717,719,7
21,727,729,731,754,755,756,757,766,767,768,769,770,771,772,782,825,826,827,828
,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847);
    -- 275 PRODUCTS
    TYPE PROD_ID_TYPE IS VARRAY(275) OF INTEGER;
    V_PROD_IDS PROD_ID_TYPE := PROD_ID_TYPE
(1750,1755,1761,1763,1768,1769,1770,1772,1774,1775,1778,1779,1780,1781,1782,17
87,1788,1791,1792,1794,1797,1799,1801,1803,1804,1805,1806,1808,1820,1822,1825,
1910,1912,1940,1948,2004,2005,2030,2049,2056,2058,2091,2093,2144,2152,2211,223
1,2236,2243,2245,2252,2253,2254,2255,2257,2259,2260,2261,2262,2264,2266,2268,2
270,2272,2274,2276,2278,2289,2293,2299,2302,2308,2311,2316,2319,2322,2323,2326
,2330,2334,2335,2336,2337,2339,2340,2350,2351,2359,2365,2370,2371,2373,2374,23
75,2377,2378,2380,2381,2382,2384,2387,2394,2395,2396,2400,2402,2403,2404,2406,
2408,2409,2410,2411,2412,2414,2415,2416,2417,2418,2419,2422,2423,2424,2430,243
9,2449,2452,2453,2457,2459,2462,2464,2467,2468,2470,2471,2492,2493,2494,2496,2
497,2522,2536,2537,2594,2596,2631,2638,2721,2722,2725,2751,2752,2761,2779,2782
,2783,2808,2810,2870,2878,2879,2944,2976,2982,2986,2995,2999,3000,3001,3003,30
04,3020,3051,3054,3057,3060,3061,3064,3065,3069,3071,3072,3073,3077,3082,3083,
3086,3087,3088,3090,3091,3097,3099,3101,3106,3108,3110,3112,3114,3117,3123,312
4,3127,3129,3133,3134,3139,3140,3143,3150,3155,3163,3165,3167,3170,3171,3172,3
173,3175,3176,3177,3178,3179,3182,3183,3187,3189,3191,3193,3197,3204,3208,3209
,3216,3220,3224,3225,3234,3245,3246,3247,3248,3250,3251,3252,3253,3255,3256,32
57,3258,3260,3262,3277,3290,3300,3301,3331,3334,3337,3350,3353,3354,3355,3359,
3361,3362,3391,3399,3400,3501,3502,3503,3511,3515);
   BEGIN
    FOR I IN 1..P_ROWS LOOP
        SELECT SEQ_ORDER_ID.NEXTVAL INTO V_ORDER_ID FROM DUAL;
        V_ORDER_DATE := TRUNC(SYSDATE- P_DAYS );
        N := ROUND(DBMS_RANDOM.VALUE(1,2));
        IF N = 1 THEN
          V_ORDER_MODE := 'direct';
        ELSE
          V_ORDER_MODE := 'online';
        END IF;
        V_CUSTOMER_ID := ROUND(DBMS_RANDOM.VALUE(1,96));
        V_ORDER_STATUS:= ROUND(DBMS_RANDOM.VALUE(1,10));
        V_SALES_REP_ID:= ROUND(DBMS_RANDOM.VALUE(153,163));
        INSERT INTO ORDERS
(ORDER_ID,ORDER_DATE,ORDER_MODE,CUSTOMER_ID,ORDER_STATUS,
                              ORDER_TOTAL,SALES_REP_ID)
        VALUES
        ( V_ORDER_ID,
          V_ORDER_DATE,
          V_ORDER_MODE,
```

```
          V_CUST_IDS(V_CUSTOMER_ID),
          V_ORDER_STATUS,
          0,
          V_SALES_REP_ID);
        V_ORDER_TOTAL := 0;
        V_ITEMS_COUNT := ROUND(DBMS_RANDOM.VALUE(1,10));
        M := 1;
        FOR X IN 1..V_ITEMS_COUNT LOOP
          V_PRODUCT_ID:= ROUND(DBMS_RANDOM.VALUE(1,275));
          SELECT LIST_PRICE INTO V_PROD_PRICE
          FROM PRODUCT_INFORMATION
          WHERE PRODUCT_ID=V_PROD_IDS(V_PRODUCT_ID);
          V_CHANGE_PRICE := ROUND(DBMS_RANDOM.VALUE(1,4));
          -- discount (MAX 20%) may be given to 25% of entered items
          IF V_CHANGE_PRICE=4 THEN
           V_PROD_PRICE := ROUND(V_PROD_PRICE -
(ROUND(DBMS_RANDOM.VALUE(1,20))/100)*V_PROD_PRICE);
          END IF;
          V_QUANTITY:=ROUND(DBMS_RANDOM.VALUE(1,20));
          V_ORDER_TOTAL := V_ORDER_TOTAL + V_QUANTITY*V_PROD_PRICE;
          BEGIN
           INSERT INTO
ORDER_ITEMS(ORDER_ID,LINE_ITEM_ID,PRODUCT_ID,UNIT_PRICE,QUANTITY)

VALUES(V_ORDER_ID,M,V_PROD_IDS(V_PRODUCT_ID),V_PROD_PRICE,V_QUANTITY );
           M := M + 1 ;
          EXCEPTION
           WHEN OTHERS THEN
            IF UPPER(SQLERRM) LIKE '%ORDER_ITEMS_UK%' THEN
             NULL;
            ELSE
             RAISE;
            END IF;
          END ;
        END LOOP; -- X loop
        UPDATE ORDERS SET ORDER_TOTAL = V_ORDER_TOTAL WHERE ORDER_ID =
V_ORDER_ID;
      IF MOD(I,100) = 0 THEN
       COMMIT;
      END IF;
    END LOOP; -- I loop
    COMMIT;
   END INSERT_ORDERS;
   PROCEDURE SpinCPUs (P_ITERATION IN NUMBER)
   IS
    N NUMBER;
   BEGIN
    FOR I IN 1.. P_ITERATION LOOP
     -- pure CPU processing (no physical or logical read)
     N := SQRT(ROUND(DBMS_RANDOM.VALUE(1,1000)));
     -- DBMS_LOCK.SLEEP(round(DBMS_RANDOM.VALUE(0.01,0.05),2)); -- in seconds
    END LOOP;
   END SpinCPUs;
   PROCEDURE RandomQuery(P_ITERATION IN NUMBER)
   IS
    V_START NUMBER;
    V_END NUMBER;
    N NUMBER;
```

```
   BEGIN
    V_END := G_MAX_ORDER_ID;
    V_START := ROUND(DBMS_RANDOM.VALUE(G_MIN_ORDER_ID,G_MAX_ORDER_ID));
    FOR I IN 1.. P_ITERATION LOOP
     FOR R IN (SELECT O.ORDER_ID, O.ORDER_DATE, C.CUST_LAST_NAME,
O.ORDER_TOTAL
      FROM ORDERS O , CUSTOMERS C
      WHERE O.CUSTOMER_ID = C.CUSTOMER_ID
      AND ORDER_ID BETWEEN V_START AND V_END) LOOP -- just retreive data
      NULL;
     END LOOP;
     -- DBMS_LOCK.SLEEP(ROUND(DBMS_RANDOM.VALUE(0.01,3),2));
    END LOOP;
   END RandomQuery;
   PROCEDURE RandomDML(P_ITERATION IN NUMBER)
   IS
    V VARCHAR2(1);
    N NUMBER;
    V_ORDER_ID NUMBER;
    V_LINE_ITEM_ID NUMBER ;
    V_MAX_ORDERI_ID NUMBER;
    V_MIN_ORDERI_ID NUMBER;
    V_PRICE NUMBER;
    V_ORDER_LOCKED BOOLEAN := FALSE;
   BEGIN
    FOR I IN 1.. P_ITERATION LOOP
     N := ROUND(DBMS_RANDOM.VALUE(1,3));
     IF N=1 THEN
      INSERT_ORDERS(1);
     ELSIF N=2 THEN
       -- pick up an order and modify the price of one of its items
       V_ORDER_ID := ROUND(DBMS_RANDOM.VALUE(G_MIN_ORDER_ID,G_MAX_ORDER_ID));
       V_ORDER_LOCKED := FALSE;
       -- make sure the order isn't locked
       BEGIN
        SELECT 'X' INTO V FROM ORDERS WHERE ORDER_ID = V_ORDER_ID FOR UPDATE
WAIT 3;
       EXCEPTION
        WHEN OTHERS THEN
         IF SQLCODE='-30006' THEN -- the row is locked
          V_ORDER_LOCKED := TRUE;
         ELSE
          RAISE; -- if there is another non-expected error, raise it
         END IF;
       END;
      IF V_ORDER_LOCKED THEN
       NULL; -- no need to make the session blocked
      ELSE -- update the item
      BEGIN
       SELECT MAX(LINE_ITEM_ID), MIN(LINE_ITEM_ID)
        INTO V_MAX_ORDERI_ID, V_MIN_ORDERI_ID
        FROM ORDER_ITEMS WHERE ORDER_ID=V_ORDER_ID;
       V_LINE_ITEM_ID :=
ROUND(DBMS_RANDOM.VALUE(V_MIN_ORDERI_ID,V_MAX_ORDERI_ID));
       SELECT UNIT_PRICE INTO V_PRICE FROM ORDER_ITEMS
        WHERE ORDER_ID=V_ORDER_ID AND LINE_ITEM_ID=V_LINE_ITEM_ID;
       V_PRICE := V_PRICE + (ROUND(DBMS_RANDOM.VALUE(-20,20))/100)*V_PRICE;
       -- update if there is no lock on the item
```

```
       BEGIN
        SELECT 'X' INTO V FROM ORDER_ITEMS WHERE ORDER_ID=V_ORDER_ID AND
LINE_ITEM_ID=V_LINE_ITEM_ID FOR UPDATE WAIT 3;
        UPDATE ORDER_ITEMS SET UNIT_PRICE=V_PRICE
         WHERE ORDER_ID=V_ORDER_ID AND LINE_ITEM_ID=V_LINE_ITEM_ID;
        UPDATE ORDERS SET ORDER_TOTAL=ORDER_TOTAL+V_PRICE
         WHERE ORDER_ID=V_ORDER_ID;
       EXCEPTION
        WHEN OTHERS THEN
         IF SQLCODE='-30006' THEN -- the row is locked
          NULL; -- do nothing
         ELSE
          RAISE; -- if there is another non-expected error, raise it
         END IF;
       END;
      EXCEPTION
       -- if order has no items
       WHEN NO_DATA_FOUND THEN
        NULL;
      END;
      END IF; -- V_ORDER_LOCKED
     ELSIF N=3 THEN
       -- pick up an item to delete
       V_ORDER_ID := ROUND(DBMS_RANDOM.VALUE(G_MIN_ORDER_ID,G_MAX_ORDER_ID));
       SELECT MAX(LINE_ITEM_ID), MIN(LINE_ITEM_ID)
        INTO V_MAX_ORDERI_ID, V_MIN_ORDERI_ID
        FROM ORDER_ITEMS WHERE ORDER_ID=V_ORDER_ID;
       V_LINE_ITEM_ID :=
ROUND(DBMS_RANDOM.VALUE(V_MIN_ORDERI_ID,V_MAX_ORDERI_ID));
       DELETE ORDER_ITEMS WHERE ORDER_ID=V_ORDER_ID AND
LINE_ITEM_ID=V_LINE_ITEM_ID;
       -- re-order the itmes in the order
       DECLARE
         CURSOR CR IS SELECT LINE_ITEM_ID FROM ORDER_ITEMS WHERE
ORDER_ID=V_ORDER_ID ORDER BY LINE_ITEM_ID FOR UPDATE;
       BEGIN
        N:=1;
        FOR R IN CR LOOP
         UPDATE ORDER_ITEMS SET LINE_ITEM_ID = N WHERE CURRENT OF CR;
         N:=N+1;
        END LOOP;
       END ;
     END IF;
     -- DBMS_LOCK.SLEEP(ROUND(DBMS_RANDOM.VALUE(0.1,2),2));
     COMMIT;
    END LOOP;
   END RandomDML;
Begin
 INITIALIZE;
END load_generator;
/

-- initial data
exec load_generator.insert_orders(100000);

exec dbms_stats.gather_schema_stats('SA');
```

- OS side: Unix

```
-- in a folder create the following scripts
mkdir ~/scripts
cd  ~/scripts
mkdir load
cd load

-- (1) loadcpu1.sh
vi loadcpu.sh
#!/bin/bash
# apply CPU load on Oracle DB
# parameters: 1 connections, 2 Iterations
users=$1
SRVC="ora11gr2"
UNPW="sa/s"
SQLCMD="/home/oracle/scripts/load/loadcpu.sql"
x=1
y=$users
ITER=$2
while [ $x -le $y ]
    do
     sqlplus -s $UNPW@$SRVC @$SQLCMD $ITER &
     x=`expr $x + 1`
done

-- (2) loadcpu.sql
vi loadcpu.sql
begin
 sa.LOAD_GENERATOR.SpinCPUs(&1);
end;
/
exit


-- (3) loadquery.sh
vi loadquery.sh
#!/bin/bash
# apply random queries load on Oracle DB
# parameters: 1 connections, 2 Iterations, 3 rows in names
users=$1
SRVC="ora11gr2"
UNPW="sa/s"
SQLCMD="/home/oracle/scripts/load/loadquery.sql"
x=1
y=$users
ITER=$2

while [ $x -le $y ]
    do
     sqlplus -s $UNPW@$SRVC @$SQLCMD $ITER &
     x=`expr $x + 1`
done

-- (4) loadquery.sql
vi loadquery.sql
begin
 sa.LOAD_GENERATOR.RANDOMQUERY (&1);
```

```
end;
/
exit


-- (5) loaddml.sh
vi loaddml.sh
#!/bin/bash
# apply random DML load on Oracle DB
# parameters: 1 connections, 2 Iterations
users=$1
SRVC="ora11gr2"
UNPW="sa/s"
SQLCMD="/home/oracle/scripts/load/loaddml.sql"
x=1
y=$users
ITER=$2
while [ $x -le $y ]
    do
     sqlplus -s $UNPW@$SRVC @$SQLCMD $ITER &
     x=`expr $x + 1`
done

-- (6) loaddml.sql
vi loaddml.sql
begin
 sa.LOAD_GENERATOR.RANDOMDML(&1);
end;
/
exit


# make all sh files in current directory executables
for f in *.sh; do  chmod 744 $f; done
```

- OS side: Windows

```
-- in a folder create the following scripts
-- (1) loadcpu.bat
REM apply CPU load on Oracle DB
REM parameters: 1 connections, 2 Iterations

set users=%1
set SRVC=ora11g
set UNPW=sa/s
set SQLCMD=C:\TEMP\load\loadcpu.sql
set x=1
set y=%users%
set ITER=%2

for /L %%i in (1,1,%y%) do (start /d "E:\oracle\OraDB11g" sqlplus -S
%UNPW%@%SRVC% @%SQLCMD% %ITER%)


-- (2) loadcpu1.sql
begin
 hrs.LOAD_GENERATOR.SpinCPU(&1);
end;
```

```
/
exit


-- (3) loadquery.bat
REM apply random queries load on Oracle DB
REM parameters: 1 connections, 2 Iterations

set users=%1
set SRVC=ora11g
set UNPW=sa/s
set SQLCMD=C:\TEMP\load\loadquery.sql
set x=1
set y=%users%
set ITER=%2

for /L %%i in (1,1,%y%) do (start /d "E:\oracle\OraDB11g" sqlplus -S
%UNPW%@%SRVC% @%SQLCMD% %ITER%)

-- (4) loadquery.sql
begin
 hrs.LOAD_GENERATOR.RANDOMQUERY(&1);
end;
/
exit


-- (5) loaddml.bat
REM apply random DML load on Oracle DB
REM parameters: 1 connections, 2 Iterations, 3 rows in names
set users=%1
set SRVC=ora11g
set UNPW=sa/s
set SQLCMD=C:\TEMP\load\loaddml.sql
set x=1
set y=%users%
set ITER=%2

for /L %%i in (1,1,%y%) do (start /d "E:\oracle\OraDB11g" sqlplus -S
%UNPW%@%SRVC% @%SQLCMD% %ITER%)


-- (6) loaddml.sql
begin
 sa.LOAD_GENERATOR.RANDOMDML (&1);
end;
/
exit
```

Using the Load Generator Scripts

```
## Using the Load Generator #
# 5 connections 1000 iterations 100000 number of rows in NAMES table
# in Windows:
loadcpu 5 1000
loaddml 5 1000

# in Unix:
```

```
./loadcpu.sh 5 1000
```

SQL Usage Samples

Merge Command

```
/* Marge Command */
MERGE INTO products p /* Destination table
USING product_changes s /* Source table
ON (p.prod_id = s.prod_id) /* Search/join condition
WHEN MATCHED THEN UPDATE /* Update if join
 SET p.prod_list_price = s.prod_new_price
 WHERE p.prod_status <> 'EXPIRED' /* Conditional update
WHEN NOT MATCHED THEN
INSERT /* Insert if not join
 SET p.prod_list_price = s.prod_new_price
 WHERE s.prod_status <> 'EXPIRED' /* Conditional insert

-- using delete keyword with Merge command
-- only rows to update will be considered by the delete clause
MERGE INTO products p
USING product_changes s ON (p.prod_id = s.prod_id)
WHEN MATCHED THEN UPDATE
 SET p.prod_list_price = s.prod_new_price,
 p.prod_status = s.prod_new_status
 DELETE WHERE (p.prod_status = 'OLD_ITEM') -- it part of update
WHEN NOT MATCHED THEN INSERT
(prod_id, prod_list_price, prod_status)
VALUES (s.prod_id, s.prod_new_price, s.prod_new_status);
```

Multitable Inserts

```
/* Multitable Inserts */
-- unconditional
INSERT ALL
INTO target1 VALUES (product_id, customer_id, sysdate, product_quantity)
INTO target2 VALUES (product_id,sysdate,product_price,product_discount)
 SELECT s.product_id, s.customer_id, sysdate, s.product_quantity,
 s.product_price, s.product_discount
 FROM source s;

-- conditional ALL rows
-- a row can be on the two tables
INSERT ALL
WHEN product_id IN(SELECT product_id FROM primary) THEN
 INTO target1 VALUES (product_id, customer_id, sysdate, product_quantity)
WHEN product_id IN (SELECT product_id FROM secondary) THEN
 INTO target2 VALUES (product_id, sysdate, product_price, product_discount)
SELECT s.product_id, s.customer_id, sysdate, s.product_quantity,
 s.product_price, s.product_discount
 FROM source s;

-- conditional first rows
--  insert into table of first applied condition -> a row is in only one table
INSERT FIRST WHEN (sum_quantity_sold > 10 AND prod_weight_class < 5) AND
sum_quantity_sold >=1) OR (sum_quantity_sold > 5 AND prod_weight_class > 5)
```

```
THEN
  INTO large_freight_shipping VALUES
      (time_id, cust_id, prod_id, prod_weight_class, sum_quantity_sold)
  WHEN sum_amount_sold > 1000 AND sum_quantity_sold >=1 THEN
  INTO express_shipping VALUES
      (time_id, cust_id, prod_id, prod_weight_class,
       sum_amount_sold, sum_quantity_sold)
WHEN (sum_quantity_sold >=1) THEN INTO default_shipping VALUES
      (time_id, cust_id, prod_id, sum_quantity_sold)
ELSE INTO incorrect_sales_order VALUES (time_id, cust_id, prod_id)
SELECT s.time_id, s.cust_id, s.prod_id, p.prod_weight_class,
       SUM(amount_sold) AS sum_amount_sold,
       SUM(quantity_sold) AS sum_quantity_sold
FROM sales s, products p
WHERE s.prod_id = p.prod_id AND s.time_id = TRUNC(SYSDATE)
GROUP BY s.time_id, s.cust_id, s.prod_id, p.prod_weight_class;

-- Mixed Conditional and Unconditional Insert
INSERT FIRST WHEN cust_credit_limit >= 4500 THEN
  INTO customers_special VALUES (cust_id, cust_credit_limit)
  ELSE INTO customers
SELECT * FROM customers_new;
```

Parallel Insert

- For further details see: Enabling Direct-Path INSERT

```
/* Parallel loading into table */
-- must before the hint
ALTER SESSION ENABLE PARALLEL DML;
-- then use the hint
INSERT /*APPEND NOLOGGING PARALLEL */
INTO sales_data
 SELECT product_id, customer_id, TRUNC(sales_date),
 discount_rate, sales_quantity, sale_price
 FROM sales_history;
-- if triggers are there, hint won't be used
```

Deleting Duplicate Rows in a Table

```
-- delete duplicated rows in our_table based on col1, col2.. etc.
DELETE FROM our_table
WHERE rowid not in
 (SELECT MIN(rowid)
  FROM our_table
  GROUP BY column1, column2, column3...) ;
```

Adding a Primary Key to a Table and then Filling it

```
create sequence seq_OAS_BALANCE;

alter table OAS_BALANCE add ( pkid number );
```

```
declare
 n number;
 cursor crs is select rowid from OAS_BALANCE for update of pkid;
begin
 for r in crs loop
  select seq_OAS_BALANCE.nextval into n from dual  ;
  update OAS_BALANCE set pkid=n where current of crs;
 end loop;
 commit;
end;
/
```

More DBA Scripts

Compare Table Sizes in Two Databases

```
--  compare tables in current db with same tables in other db
--   useful for checking after transferring too many tables

DECLARE
 N NUMBER;
 M NUMBER;
BEGIN
 DBMS_OUTPUT.ENABLE(100000);
 FOR S IN ( SELECT USERNAME FROM SCHEMA_NAMES WHERE USERNAME <>'MOL' ORDER BY
1) LOOP
   FOR O in ( SELECT TABLE_NAME FROM DBA_TABLES@OLDORA11G WHERE OWNER =
S.USERNAME ) LOOP
    BEGIN
     N := 0;
     M := 0;
     EXECUTE IMMEDIATE 'SELECT COUNT(ROWID) FROM ' || S.USERNAME ||'."'||
O.TABLE_NAME || '"@OLDORA11G' INTO N;
     EXECUTE IMMEDIATE 'SELECT COUNT(ROWID) FROM ' || S.USERNAME ||'."'||
O.TABLE_NAME || '"' INTO M;
     IF M <> N THEN
      DBMS_OUTPUT.PUT_LINE(S.USERNAME || CHR(9) || O.TABLE_NAME);
     END IF;
    EXCEPTION
     WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE(S.USERNAME || CHR(9) || O.TABLE_NAME);
      RAISE;
    END;
   END LOOP;
 END LOOP;
END;
/
```

Using Linux for Oracle

Verifying the Kernel

```
# is of format: <Base>-<Errata>.<Mod><Type>
# Mod= Modifier (developer)
# Type: None: Uniprocessor, smp: SMP up to 16 GB, hugemem: SMP up to 64 GB
uname -r
2.6.9-42.0.0.0.1.ELsmp

# name of the release
cat /etc/enterprise-release
```

Checking for a Tainted Kernel

- Usually, tainted kernelt is not supported by Oracle. Some exceptions: OCFS (276450.1) and EMC Powerpath driver (284823.1)

```
# 0: not tainted  1:tainted
cat /proc/sys/kernel/tainted

# to list the status of modules
lsmod
```

Supported Hardware

- Systems listed in Red Hat site (https://hardware.redhat.com) are supported by Oracle Linux.

Using Oracle Relink Utility

- If the operating system or Oracle object files change, then Oracle executables should be relinked to create updated executables properly integrated with the operating system.
- If utility returns nothing, then it means it successfully finished.
- Relinking occurs automatically when a patch set is installed, but must be performed manually when:
  - An OS upgrade has occurred
  - A change has been made to the OS system libraries. This can occur during the application of an OS patch.
  - A new installation failed during the relinking phase.
  - An individual Oracle patch has been applied. Explicit relink instructions are usually either included in README or integrated into the patch install script.

```
1. Verify whether ORACLE_HOME is correctly set.
2. Set LD_LIBRARY_PATH to include $ORACLE_HOME/lib.
3. Execute the appropriate relink command to relink the
appropriate binaries:
– relink all: All Oracle executables
– relink oracle Database executables
– relink network: Listener, cman, names
```

```
– relink oemagent: Intelligent Agent
– relink utilities: SQL*Loader, tkprof, rman,
impdp, expdp, imp, exp
– relink ctx: Oracle Text
```

Certified and Supported File Systems

- In Linux, Oracle certifies three file systems:
  - ext3
  - Automatic Storage Management (ASM)
  - Oracle Cluster File System/Oracle Cluster File System 2 (OCFS2)

Enterprise Linux Runlevels

- Runlevel 0: Halt. It is used to shut down the system.
- Runlevel 1: Single-user (maintenance) mode. Only root may log in.
- Runlevel 2: Multiuser mode, text-based console only
- Runlevel 3: Multiuser mode with networking
- Runlevel 4: For custom use
- Runlevel 5: The default mode; multiuser with networking and active X session
- Runlevel 6: Reboot. This shuts everything down and then starts it back.

```
Booting to a Nondefault Runlevel
Enterprise Linux:
>you press [E] then scroll down to select the kernel
>press [E] again
>press the spacebar
>enter the runlevel you want to boot to
>then press [B]

# to print previous and current runlevels:
runlevel
```

Using /etc/oratab File and dbstart Utility

- Y or N in /etc/oratab indicates to the dbstart utility whether the database should be brought up at system boot time.
- ASM and database instances created without using DBCA will not appear in the oratab file. You should manually add them:

```
+ASM:/u01/app/oracle/product/10.2.0/db_1:Y
orcl:/u01/app/oracle/product/10.2.0/db_1:Y
```

```
The dbstart script that is provided with 10.2.0.1.0 contains an error.
The Metalink note 336299.1 documents this issue. To solve this problem,
perform the following steps:
1. Edit $ORACLE_HOME/bin/dbstart
2. Go to line 78.
3. Replace the line with:
```

```
ORACLE_HOME_LISTNER=$ORACLE_HOME
4. Save the file and exit the editor.
5. Execute dbstart.
```

Automating Jobs

Using cron

```
/* security */
Only users listed in the /etc/cron.allow file are allowed to use cron
Users listed in the /etc/cron.deny file cannot use cron.
If neither of these files exist, only the superuser can use cron.


/* Managing crontab files */
crontab  used  to install, deinstall or list the tables (crontabs).
-u user the user whose crontab is to be tweaked
-l      display the current crontab file
-r      remove the current crontab file
-e      edit. After exit from the editor, the modified crontab will be
installed


/* Using cron */
crontab format: <minute> <hour> <day> <month> <weekday> <cmd>
1.minute (from 0 to 59) or special keywords:
2.hour (from 0 to 23)
3.day of month (from 1 to 31)
4.month (from 1 to 12)
5.day of week (from 0 to 6) (0=Sunday)

Special keywords are:
@reboot     Run once, at startup
@yearly     Run once  a year     "0 0 1 1 *"
@annually   (same as  @yearly)
@monthly    Run once  a month    "0 0 1 * *"
@weekly     Run once  a week     "0 0 * * 0"
@daily      Run once  a day      "0 0 * * *"
@midnight   (same as  @daily)
@hourly     Run once  an hour    "0 * * * *"

If the administrator puts an executable script into one of the following
directories,
then cron runs the script at the appropriate interval:
/etc/cron.hourly
/etc/cron.daily
/etc/cron.weekly
/etc/cron.monthly


/* Mailing the crontab output */
By default cron saves the output in the user's mailbox on the local system.
But you can also configure crontab to forward all output to a real email
address:
crontab
MAILTO="yourname@yourdomain.com"
```

```
If want to receive only one cronjob's output in your mail,
make sure this package is installed:
aptitude install mailx
then change the cronjob like this:
*/10 * * * * /bin/execute/this/script.sh 2>&1 | mail -s "Cronjob ouput"
yourname@yourdomain


/* Examples */
>every Friday 1AM
0 1 * * 5 /bin/execute/this/script.sh

>14 minutes after 10 p.m. every Monday through Friday of every month
14 22 * * 1-5 su - oracle -c /usr/local/bin/backup.cmd >/dev/null 2>&1

>10 past after every hour on the 1st of every month
10 * 1 * * /bin/execute/this/script.sh

>every 10 minutes
0,10,20,30,40,50 * * * * /bin/execute/this/script.sh
*/10 * * * * /bin/execute/this/script.sh

>daily
@daily /bin/execute/this/script.sh
```

Using anacron

- Most system cron jobs will run during the night and will get done only if the computer is switched on. The anacron mechanism on the other hand assumes that the computer is not running continuously. After the computer is started anacron will execute jobs after a certain delay.
- anacron keeps track of the day a given job has been executed. So the shortest period a job can be executed is a day.
- The anacron daemon is usually started from an rc-script and will read an anacrontab which has the following format:

  `Period  Delay  Job-Identifier  Command`
- The period is given in days, the delay is in minutes, the job-identifier can be any character and the command is usually of the form `run-parts /etc/cron.daily` but can be any command.

```
Example 9.4. Sample /etc/anacrontab
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
1     65    cron.daily       run-parts /etc/cron.daily
7     70    cron.weekly      run-parts /etc/cron.weekly
30    75    cron.monthly     run-parts /etc/cron.monthly
```

Using at command

- The at command executes a command once at a specified time.
- By default at is restricted to the root user. To override this you must either have an empty /etc/at.deny or have a /etc/at.allow with the appropriate names.
- at jobs are saved in /var/spool/at/

- at jobs are spooled in /var/spool/at/

```
at [time]
Where time can be expressed as:
now
now + 1 min
now + 1 hour
now + 1 day
3am + 2days
midnight
10:15 Apr 12
teatime
For complete list of valid time formats see /usr/share/doc/at-xxx/timespec

# execute the commands in the file
at -f myjobs.txt now + 1 hour

# list commands that are scheduled
at -l
atq

# remove a job
atrm <jobid>
```

Using batch command

- batch command executes commands when system load average drops below 0.8, or the value specified in the invocation of atrun.

```
batch [-V] [-q queue] [-f file] [-mv] [TIME]
```

Task Scheduler

- There are GUI task schedulers available that are front ends for cron in both the Gnome and the K(ommon) Desktop Environment (KDE).

Configuring Linux Memory for Oracle

- 32 bit Architecture:
- With Linux 32-bit, Oracle is able to use about 1.7GB of address space for its SGA. To increase it, refere to noted 200266.1. Be aware of the consequenced limitations.
- In Red Hat Enterprise Linux 3, 4 or 5 the smp kernel can be used on systems with up to 16 GB of RAM.
- The hugemem kernel is required in order to use all the memory on systems that have more than 16GB of RAM up to 64GB. However, it is recommend to use the hugemem kernel even on systems that have 8GB of RAM.
- In Red Hat Enterprise Linux 5, a 32 bit kernel is always a hugemem kernel so there is no need to install a special kernel.

Using Linux Performance Monitoring Tools

Note: OS Watcher uses most commonly used Linux tools and store their output in files. It is a nice frame work to implement using those tools. To learn more about using it, see Using OS Watcher.

About Linux Tools

- top: useful for diagnosing CPU and I/O issues, though not as reliable for Oracle memory issues. Individual process memory numbers should not be relied upon.
- sar: reports system activity including memory, I/O, and CPU usage.

  sar samples the /proc file system at 10-minute intervals and records the data in /var/log/sa as a sa# file where # is the day of the month. A full month's worth of statistics are retained so that the administrator can use sar to view past statistics as well as current. For example, to view the CPU statistics for the 23rd of the month:

  sar –u 2 4 –f /var/log/sa/sa29

  -o [<fnmae>] save output to filename (default /var/log/sa/sadd) (cannot be used with –X)

  -f [<fnmae>] read from filename (default /var/log/sa/sadd)

  -X pid | SELF | ALL Report statistics for the child processes of the process whose PID is pid.

  SELF statistics of the child processes of the sar process itself

  ALL statistics of the child processes of all the system processes.
- vmstat: activity on processes, memory, paging, block I/O, traps, and CPU activity (best for memory)
- iostat: disk activity
- System Log Files: /var/log/message ../syslog
- free command displays the total amount of free and used physical and swap memory in the system, as well as the shared memory and buffers used by the kernel.
- mpstat: reporter on CPU statistics.
- Graphical Measurement Tools

Using Linux Tools

```
/* CPU */
# how many CPUs
cat /proc/cpuinfo

# use top to see averae load (now, 5 minutes ago, and 10 minutes ago).
# load factor= load average/#CPUs
# if lf<1: no load on CPU,
#      between 1 and 2: CPU running at capacity
#      >2, CPU may be a bottleneck (use sar and vmstat to investigate)
# if you use "1" interactive command, you can see load per CPU
top

# CPU metrics
# all appended by <interval> <count>
vmstat
mpstat -P <CPU>|ALL
sar -u
# queue size: waiting for CPU
# (shouldn't be larger than CPU#)
sar -q
# display CPU stats only
iostat -c

# to know which session is consuming CPU
```

```
# get its PID from top
# then:
select pid, spid, pname, username, program
from v$process
where spid=20052;

select * from v$session
where paddr= ( select addr from v$process where spid=20304);


/* Memory */
# Total Memory
free -m, top, or cat /proc/meminfo

# higher "Pages In" indicate RAM shortage
top
free
cat /proc/meminfo
# paging info
sar -B
# memory metrics
sar -R


/* IO */
# storage vendors have their monitoring tools
# io activity by disk
# check avgrq-sz.await which gives the average wait time for requests
iostat -d
# by partition
iostat -p
# exetended output
iostat -d -x
# rtps/wtps requests per se
sar -d 2 10
# about overall io stats
vmstat
```

Checking Some General Guideline on Truning Oracle in Linux

- Use an appropriately sized database block: general purpose:8K, OLTP:4K, DSS:16K. Use a larger block size if the application has *very large* indexes (15>levels), or consider using partitioned indexes.
- Use a larger redo log buffer than default: for OLTP systems
- Multiprocessing (SMP) machines.
- Setting PRE_PAGE_SGA = true causes each server process that starts to touch every page of the SGA, thus mapping all the SGA pages to physical memory. It may slow down connection time, though; so make sure you have enough free memory to accomodate all SGA.
- Use automatic memory management.
- Setting atime off: In Linux, by default, the last time-read attribute (atime) is updated every time a file is read. For database files, this is not an important statistic and you can turn it off to reduce the number of I/Os:

```
# This parameter can be set for a file with:
chattr +A <filename>
# for a directory with:
chattr -R +A <directory name>

# To make it persistent across reboots:
vi /etc/fstab file
/dev/hdb7 /u3/app/oracle/ ext3 rw,noatime 1 1
```

Troubleshooting Oracle Database in Linux

Using OS Watcher (OSW)

- OS Watcher (OSW) is a collection of UNIX shell scripts intended to collect and archive operating system and network metrics to aid support in diagnosing performance issues. OSW operates as a set of background processes on the server and gathers OS data on a regular basis, invoking such Unix utilities as vmstat, netstat and iostat.
- OSW will need access to the OS utilities: top, vmstat, iostat, mpstat, netstat, and traceroute.
- On RAC, you can install it in every node.
- After generating the output files, consider generating a profile from oswg (next section).
- Obtain latest version from Note 301137.1

```
/* Managing OSW */
# installing OSW (directory osw will be created):
tar xvf osw.tar


# Uninstalling OSW
rm -rf osw


# Setting up OSW for tracing private networks (RAC):
mv Exampleprivate.net private.net


# starting OSW
./startOSW.sh <interval in seconds> <hour# retention>
# to startup in background:
nohup ./startOSW.sh 30 12 &


# Stopping OSW
./stopOSW.sh



/* Diagnostic Data Output */
# archived files generated per hour with the following format:
<node_name>_<OS_utility>_YY.MM.DD.HH24.dat


# there are 3 entries for each timestamp. You should always ignore
# the first entry as this entry is always invalid. The second and
```

```
# third entry will be valid but the second entry will be 1 sec later
# than the timestamp and the third entry will be 2 seconds later than the
timestamp.


# oswiostat
# What to look for:
- Average service times greater than 20msec for long duration.
- High average wait times
r/s     Shows the number of reads/second
w/s     Shows the number of writes/second
kr/s    Shows the number of kilobytes read/second
kw/s    Shows the number of kilobytes written/second
wait    Average number of transactions waiting for service (queue length)
actv    Average number of transactions actively being serviced
wsvc_t  Average service time in wait queue, in milliseconds
asvc_t  Average service time of active transactions, in milliseconds
%w      Percent of time there are transactions waiting for service
%b      Percent of time the disk is busy
device  Device name


# oswmpstat
What to look for
- Involuntary context switches (this is probably the more relevant statistic
when examining performance issues.)
- Number of times a CPU failed to obtain a mutex. Values consistently greater
than 200 per CPU causes system time to increase.
- xcal is very important, show processor migration


# oswnetstat
# the collisions should be 0.1 percent or less
Network collision rate = Output collision / Output packets
# Input Error Rate = Ierrs / Ipkts
If the input error rate is high (over 0.25 percent), the host is excessively
dropping packets.
# segment retransmission rate should be low
%segment-retrans=(tcpRetransSegs / tcpOutDataSegs) * 100


# oswprvtnet
check Interface is up and responding


# oswps
list of processes


# oswtop
# examine average load
# Large run queue
# A process which is "hogging" CPU is always suspect
# number of processes usually do not change over time


# oswvmstat
```

Using OS Watcher Graphs(OSWg)

```
Using OSWg

# to use Java shipped with Oracle
$ cd $ORACLE_HOME/jre/1.4.2/bin
export PATH=/u01/app/oracle/product/11.2.0/db_1/jdk/jre/bin:$PATH


# invoking the oswg:
java -jar oswg.jar -i /home/oracle/osw/archive
# if you see: java.lang.OutOfMemoryError,
#  you may have to increase the size of the java heap:
$java -jar -Xmx512M oswg.jar -i /home/oracle/osw/archive

Note: to generate a profile, mkdir profile under osw home

# Using OSWg: Menu Option
java -jar OSWg.jar -i <fully qualified path name of an osw archive directory>

# other starting options:
java -jar OSWg.jar -i <fully qualified path name of an osw archive directory>
-P <name> -L <name> -6 -7 -8 -B <time> -E <time>

-B <start time> Same as option T from the menu. The start time will allow the
user to select a start time from within the archive of files to graph/profile.
This overrides the default start time which is the earliest time entry in the
archive directory.  The format of the start time is  Mon DD HH:MM:SS YYYY.
(Example :Jul 25 11:58:01 2007). An end time is required if selecting this
option.
-E <end time> Same as option T from the menu. The end time will allow the user
to select an end time from within the archive of files to graph/profile. This
overrides the default end time which is the latest time entry in the archive
directory.  The format of the end time is  Mon DD HH:MM:SS YYYY. (Example :Jul
25 11:58:01 2007). A start time is required if selecting this option.
```

Using the On-Board Monitor (LTOM)

- LTOM is an embedded real-time data Collection and diagnostics platform developed by Center of Expertise in Oracle.
- LTOMg is embedded in LTOM.
- Its reference is Note 352363.1.
- Usage details are in README file.
- Automatic Hang Detection
  - it starts collecting statisics about hanged sessions
  - to use it for RAC, install LTOM in a single node. (Other features require installing it in all nodes)
  - uses a rule based hang detection algorithm configured in $TOM_HOME/init/hangDetect.properties
- System Profiler
  - The System Profiler provides the ability to continually collect data from both the operating system and oracle and provides an integrated snapshot of the overall health of the operating system together with the database.
  - Once the data is collected, the data can be parsed and analyzed through LTOMg.

- Automatic Session Tracing
  - Automatic Session Tracing uses a set of rules to determine when to turn on SQL trace for individual oracle sessions, using event 10046 level 12 trace.

```
/* installing ltom */
# Running LTOM
# os user must be member of dba group
# db user required with full dba priv
tar -xf ltom420.tar
# the script creates "tom" user which can be dropped after installation
cd ltom/tom_base/install
# if default java cannot help, use Oracle's one (add it to .bash_profile)
export PATH=/u01/app/oracle/product/11.2.0/db_1/jdk/jre/bin:$PATH
./autoinstall.sh


# uninstall
rm -rf tom_base


/* TOOL USAGE */
# insertactive
export TOM_HOME=/home/oracle/ltom/tom_base/tom
cd /home/oracle/ltom/tom_base/tom
./startltom.sh

# in the background:
see the README file

# running ltomg
java -jar LTOMg.jar -i
/home/oracle/ltom/tom_base/tom/recordings/profile/pro1284214950770.log
java -jar -Xmx512M LTOMg.jar -i
/u02/home/ltom/recordings/profile/pro1190754096675.log
```

Using strace

- Strace is a utility that intercepts and records the system calls, which are called by a process, and the signals, which are received by a process.

```
strace –o /tmp/helloworld.out ./helloworld.sh

strace –aef –Ttt –o /tmp/date.out date

strace –p 1287 –o /tmp/ora_pmon_orcl.out
```