

DB Tools for OracleTM 6.1

User's Guide

Supported database systems: Oracle 7.3, 8.0, 8i, 9i, 10g, and 11g

This page has been intentionally left blank

Contents

About this guide	18
Intended Audience	18
Conventions used in this document.....	18
Abbreviations and Product Reference Terms.....	18
Trademarks	18
DB Tools Quick Start Menu	19
Common Dialogs and Controls.....	21
Application Toolbar.....	21
Status Bar.....	21
Connect to Database Dialog.....	21
Sort Dialog.....	23
Crosstab Definition Dialog	24
Updating Crosstab Definition	24
Changing the names used for the columns and rows	24
Filter Dialog	26
Modify Expression Dialog.....	27
Zoom Dialog	27
Search Dialog.....	28
Row Descriptor Control	28
Split Control.....	28
Sampling Cycle	29
Schema Names Cache.....	30
Schema Selection Dialog	30
Working with data and import/export	30
Sorting	30
Filtering.....	31
Navigating.....	31
Interrupting data retrieval.....	31
Printing	31
Selecting text in a modifiable field.....	32
Selecting fields, columns, rows in a datasheet	32
Copying	33
Editing.....	33
Saving.....	34
Exporting data	34
Importing data.....	35
Searching	35
Zooming in/out.....	36
Resizing columns and rows	36
Navigation Buttons.....	36
Customizing DB Tools Interface	36
Timer Options	37
Database Operations Options	37
Editor Options	38
Print Options	38
DDL Export Options.....	39
General Options.....	39

Printing	40
Print Options	40
Window Menu Commands	41
Table painter and table/view graphical design.....	42
Working with objects in the Table Painter workspace.....	42
Opening a table.....	42
Graphically selecting data from a table	43
Moving objects	43
Resizing objects.....	43
System menu	43
Object popup menu.....	43
Modifying table properties	45
Opening related tables	45
Closing a table	45
Keyboard alternatives	46
Select Table(s) dialog.....	47
Table Design	47
Overview.....	47
Table Design dialog elements	48
Primary Key Design	50
Foreign Key Browser	50
Foreign Key Dialog	51
DDL Reverse-Engineering.....	51
DDL Reverse-Engineering Options.....	52
Reverse-engineering DDL Using Popup Menu.....	53
View Design Dialog.....	53
To create new database view	53
To alter existing database view	53
Functions in Sort and Filter Expressions	54
Functions by category.....	54
Aggregate functions	54
Crosstab functions	54
Data type checking and conversion functions	55
Date and time functions	55
Miscellaneous functions	55
Numeric functions	56
String functions	56
Asc	56
Char	57
Integer	57
IsDate.....	57
IsNull	57
IsNumber	58
IsTime	58
Long.....	58
Number	59
GetRow	59
IsRowModified.....	59
IsRowNew.....	59
Date	59
Time.....	60
DateTime	60

Day.....	61
DayName	61
DayNumber.....	61
DaysAfter	61
Hour	62
Minute	62
Month.....	62
Now.....	62
RelativeDate	63
RelativeTime	63
Second.....	63
SecondsAfter.....	63
Today	64
Year	64
Case.....	64
If.....	65
ProfileInt.....	65
ProfileString	66
RGB	66
Abs.....	67
Ceiling	67
Cos.....	67
Sin.....	68
Exp.....	68
Fact.....	68
Int.....	68
Log	69
LogTen.....	69
Mod.....	69
Pi.....	70
Rand	70
Round	70
Sign.....	70
Sqrt	71
Tan.....	71
Truncate.....	71
Fill	72
Left	72
RightTrim	72
LeftTrim.....	72
Len	73
Lower	73
Upper	73
Match	73
Mid	75
Pos.....	76
Replace.....	76
Right.....	77
Space.....	77
String.....	77
Format Symbols.....	78
Trim.....	80

WordCap	80
CrosstabAvg	80
CrosstabCount	81
CrosstabMax	81
CrosstabMin	82
CrosstabSum	82
Operators in Sort and Filter Expressions	83
Arithmetic operators	83
Relational and logical operators	83
Concatenation operator	84
Operator precedence in expressions	84
Hot Keys and Shortcuts	84
SQL Editor and DBA Notepad	85
Installation and Uninstallation	87
Installation	87
Uninstallation	87
Front-end:	87
Back-end:	87
System Requirements	87
Minimum Hardware Requirements	87
Software Requirements	88
The Dynamic Performance Tables	88
Portability	88
Connecting to Oracle databases	89
Required Oracle Client Software	89
What is SQL*Net?	89
What you must install?	89
SQL*Net	89
Network Adapter Layer	89
Testing Connection Outside of DB Tools	89
How DB Tools Talks to Oracle	89
Overview	89
Connect to Database Dialog	89
Common Problems	90
Missing network module	90
Path errors	90
Missing Required Support Files	90
Improperly Installed Required Support Files	90
Incorrect server parameter in transaction Object	90
Debugging	90
Special Considerations with Oracle 7.3	90
Special Considerations with IPX / SPX	90
Special Considerations for NT 4.0 or above	91
Common Error Messages	91
Connecting to Oracle using Oracle ODBC Drivers	92
Configure the ODBC Data Source	92
Verify Your Oracle ODBC Connection	92
Common Connect Errors	92
Creating ODBC data source for Oracle database	92
DB Monitor Expert	94
Overview	94

Why you have to monitor your database regularly	94
Active Desktop	95
Performance Graphs (Graphs menu)	95
Performance Summary Graphs	95
Real-time Statistics Graphs.....	95
Recent Statistics Graphs.....	96
Historical Statistics Graphs	96
Database Actions Graph.....	96
SQL Activity Graph	96
SGA Memory Usage Graph.....	96
License/Users Logged On Graph	96
Rollback Segment (Gets/Writes) Graph.....	96
Rollback Segment (Hits) Graph	96
Rollback Segment Transactions Graph	96
Redo Log Latch (Hits) Graph.....	97
Redo Log Latch (Misses) Graph.....	97
CPU Usage Graph.....	97
Memory Usage Graph	97
Buffer Cache Graph.....	97
Dictionary Cache Graph	97
Recursive Calls Graph.....	97
Sorts Graph	97
Waits Graph.....	97
File Input/Output (Hits) Graph.....	97
File Input/Output (Blocks) Graph	97
Custom Graph	98
Exporting Graphs	99
Background Database Performance Statistics Collection Job.....	99
Monitor connections and resources (Monitor menu)	104
Connections.....	104
Open Cursors	104
SQL Area.....	105
Object Access.....	105
Cached Objects	105
Application Effectiveness.....	105
File Input/Output	105
Rollback Segments.....	105
Rollback Transactions	106
Buffer Cache.....	106
Dictionary Cache	106
Redo Log Latches.....	106
Recursive Calls.....	106
Dispatcher Processes.....	106
Sorts	106
Waits.....	106
Locks	107
SGA.....	107
Custom Monitors.....	107
Using Thresholds in Custom Performance Monitors.....	109
Sending Messages to Pagers	109
Sending E-Mail Messages	111
Sending Messages to Beepers	112

Performance Library	112
Enhanced Explain Plan	112
Object analysis tools	113
Table Analyzer	113
Index Analyzer	114
Cluster Analyzer	115
Scheduler	116
Scheduling a new job:.....	116
Connecting scheduler to the database.	117
Locking DB Monitor Expert console when you are not around.	117
To Stop a Job:	118
Connection Manager	118
Index Histograms	119
Data Histograms.....	120
Reports.....	121
Objects Report	121
Tables Report.....	121
Indexes Report.....	121
Clusters Report	122
Tablespaces Report	122
Database Files Report	122
Sequences Report	122
Snapshots Report	122
Synonyms Report.....	122
Database Links Report.....	122
Users Report.....	122
Profiles Report	122
Table Privileges Report.....	122
Triggers Report	122
Procedures, Functions And Packages Report	123
Views Report.....	123
Fragmentation Report	123
SGA Report.....	123
Parameters Report.....	123
Version Report	123
Custom Reports.....	123
Choosing a presentation style	125
Status Analyzer	125
Dependencies Analyzer	126
DB Trends Expert	128
DB Trends - Installation/Uninstallation	128
Upgrading DB Trends package	128
Disabling DB Trends data collections	128
Uninstalling DB Trends package.....	128
Historical Trends	129
Table Trends	129
Index Trends.....	130
Cluster Trends.....	130
Schema Trends	130
Tablespace Trends.....	130
File Trends.....	131
Rollback Segment Trends	131

Database Trends	131
Custom Trend Graph	132
Selecting Objects and Spaces for Automated Trend Data Collections	133
DB Difference Expert.....	134
Overview	134
Before You Begin	134
Comparison Options.....	134
How to Analyze	135
Understanding and Using Comparison Results.....	136
DB Index Expert.....	137
Overview	137
Command Line Parameters.....	137
Analysis Options.....	138
Analysis Options for Indexes	138
Analysis Options for Schemas.....	139
How to Analyze.....	140
Interactive mode.....	140
Batch mode	140
Reports.....	140
Index Analysis Report.....	140
Referential Constraints Analysis Report	140
Library Cache Analysis Reports	141
Full Table Scans Report.....	141
Used Indexes Report	141
Detailed Library Cache Analysis Report.....	141
Report Formats.....	141
DB Tuning Expert	142
Overview	142
What DB Tuning Expert Can Do For You	142
Before You Begin.....	142
How To Tune Your Database	142
Tuning methodology	143
Performance Analysis Options	143
Report Output Formats	143
DB Space Expert.....	145
Overview	145
Using DB Space Expert's Interface	145
Tablespace Navigator.....	145
Starting Tablespace Navigator	146
Navigator Views and Properties	146
Context Menus	147
Working with Tablespaces and Files	147
Tablespace Maps	147
Overview	147
Using the Tablespace Map utility	148
Creating New Tablespaces.....	148
Taking Tablespaces Off-line/On-line.....	150
Dropping Tablespaces.....	151
Adding Files to Tablespaces.....	151
Resizing Tablespaces.....	152

Moving and Renaming Database Files	152
Converting Tablespaces	153
Modifying Default Storage Parameters	154
Managing User Space Quotas and Defaults	154
Coalescing Tablespace Free Space	155
Estimating Total Unused Space in All Objects	156
Freeing Unused Space in All Objects	156
Emptying Tablespace Recycle Bin	157
Reorganizing and Defragging Tablespaces	157
Rebuilding Tablespaces	157
Moving Tablespace Objects to Another Tablespace	158
Working with Schema Objects	158
Estimating Table and Index Space Requirements	158
Dropping Schema Objects	159
Moving Schema Objects Between Tablespaces	159
Rebuilding Indexes	159
Freeing Object's Unused Space	159
Fixing Chained Rows	160
Working with Partitions	160
Converting Regular Tables to Tables with Partitions	160
Converting Partitioned Tables to Regular Tables	163
Renaming Partitions and Subpartitions	164
Splitting Partitions	164
Merging Partitions	167
Exchanging Partition and Table Data	168
Rebuilding Index Partitions and Subpartition	169
Rebuilding Unusable Indexes	169
Reorganize / Defrag Tablespace Wizard	170
Top-level reorganization options	170
Batch Methods for Separating Objects of Different Types	171
Objects Defragging and Free Space Defragging Strategies	172
Error Handling Methods	172
Advanced Processing Options	173
Reorganization Job Scheduling	174
Rebuild Tablespace Wizard	174
Top-level reorganization options	174
Error Handling Methods	175
Advanced Processing Options	176
Reorganization Job Scheduling	176
Move Objects Wizard	176
Using Move Objects Wizard	177
Reorganization Job Scheduling	178
Fix Chained Rows Wizard	178
Using Fix Chained Rows Wizard	179
Reorganization Job Scheduling	180
Working with Database Recycle Bin	180
Running Space Reorganization Jobs	181
Job Execution Methods	181
Job Logging	182
Advantages and disadvantages of different reorganization job types	182
Monitoring and Managing Space Reorganization Jobs	182
Reports	184

DB Consumer Expert	186
Overview	186
Managing Database Users	186
User Properties	187
Enabling and Disabling User Accounts	188
Forcing Users to Change Their Passwords	188
Granting and Revoking Roles	188
Granting and Revoking Consumer Groups	189
Granting and Revoking System Privileges	190
Granting and Revoking Object Privileges	191
Creating New Database Users	193
Deleting Database Users	193
Managing Database Roles	193
Role Properties	194
Granting and Revoking Roles to Users	195
Granting and Revoking Roles to Roles	195
Granting and Revoking Consumer Groups	195
Granting and Revoking System Privileges	195
Granting and Revoking Object Privileges	195
Creating New Database Roles	196
Deleting Database Roles	196
Managing Database Profiles	196
Profile Properties	197
Working with Profiles	198
Managing Space Usage Quotas	200
Managing Resource Usage Quotas	201
Working with Resource Quotas and Profiles	203
Resource Usage Limits	203
Connect & Idle Time Limits	203
CPU Usage Limits	203
Input/Output Limits	203
Memory Usage Limits	204
Composite LimitsError! Bookmark not defined.Error! Bookmark not defined.Error! Bookmark not defined.	204
Resource Usage Plans	205
Managing Consumer Groups	205
Consumer Group Properties	206
Assigning Users and Sessions to Consumer Groups	206
Static Assignments	207
Dynamic Assignments	208
Creating New Consumer Groups	209
Deleting Consumer Groups	209
Managing Resource Usage Plans	210
Resource Plan Properties	210
Linking Consumer Groups to Resource Plans	211
Assigning CPU Service Levels to Consumer Groups	212
Limiting Resources Available to Consumer Groups	214
Dynamic Switching Between Consumer Groups	215
Time-based Resource Plan Activation	216
Enabling/Disabling Database Resource Manager	217
Creating New Resource Plans	217
Deleting Resource Plans	217

Security Navigator	217
Reports	218
Users Report	220
Profiles Report	220
Space Usage by User Report	220
Users Exhausting Space Quotas Report	220
Top 10 Users by Space Usage	221
Top 10 Users by CPU Usage	221
Top 10 Users by Memory Usage	221
Top 10 Users by Disk I/O (Input/Output)	221
Space Usage Trends	222
CPU Usage Trends	222
Memory Usage Trends	222
I/O Usage Trends	222
Application Resource Consumption Report	222
Application Performance and Trends Reports	223
Database Load Balancing Report	223
Top 10 Applications by CPU Usage	223
Top 10 Applications by Memory Usage	224
Top 10 Applications by Sorts	224
Top 10 Applications by Disk I/O (Input/Output)	224
DB Security Expert	225
Overview	225
Analyzing Effective Security Settings	225
Managing Database Users	226
User Properties	226
Enabling and Disabling User Accounts	227
Forcing Users to Change Their Passwords	228
Granting and Revoking Roles	228
Granting and Revoking Consumer Groups	229
Granting and Revoking System Privileges	229
Granting and Revoking Object Privileges	230
Creating New Database Users	232
Deleting Database Users	232
Managing Database Roles	232
Role Properties	233
Granting and Revoking Roles to Users	234
Granting and Revoking Roles to Roles	234
Granting and Revoking Consumer Groups	234
Granting and Revoking System Privileges	234
Granting and Revoking Object Privileges	234
Creating New Database Roles	235
Deleting Database Roles	235
Managing Database Profiles	235
Profile Properties	236
Working with Profiles	237
DB Log Expert	240
Overview	240
Redo Log Configuration and Status	240
Redo Log Latches	240
Database Load History Report	240

Database Load Weekly Analysis – Day by Day Comparison	241
Checkpoints Monitor	241
Database Load Trends Report	241
DB Alert Expert	242
Overview	242
DB Alert Installation and Usage.....	243
Configuring DB Alert.....	243
Databases	244
Communication Methods	245
Alert Actions	246
Log Monitor Options	247
Space Monitor Options.....	247
Database Error Monitor Options.....	248
Starting DB Alert on Computer Startup	249
Monitoring Alert Log Files and Trace Files	250
Monitoring Free Space in Tablespaces	250
Monitoring Database Errors.....	251
DB Audit Expert	252
Overview	252
System Auditing Considerations.....	252
Before You Begin System Auditing	253
Archiving system audit trail to a table	253
Archiving system audit trail to a file	253
Purging system audit trail	253
System Audit Options	253
Advanced System Auditing Options	254
Automatic Audit Trail Space Management.....	254
Audit Method and Storage	256
Auditing Privileged Users connected as SYSDBA or SYSOPER	257
Auditing Oracle Database Errors	259
Data Change Auditing	261
Before You Begin Data Auditing	261
Privileges required.....	261
Archiving data audit trail to a table.....	262
Archiving data trail to a file.....	262
Purging data audit trail:.....	262
Enabling Data Change Auditing.....	262
Disabling Data Change Auditing.....	263
DB Patch Expert.....	264
Overview	264
Patch Analysis and Reporting	264
Patch Summary Report	264
Installed Patches Report.....	265
Missing Critical Patches Report.....	265
Missing Critical and High Severity Patches Report.....	265
Missing Critical, High, and Medium Severity Patches.....	266
All Missing Patches.....	266
Automated Alerts and Notifications	266
Patch Management Tasks.....	267
DB Job Expert.....	268

Overview	268
Job Scheduler Console	269
Navigation techniques	269
Creating a new job	269
Modifying properties of an existing job	270
Toggling job disabled/enable state	271
Changing job privileges	271
Removing an existing job	272
Running a job on demand	272
Stopping a running job	272
Retrieving job log and statistics	272
Job Monitor Installation	273
Job Performance Statistics and Job Monitor Options	273
Job Performance Statistics	273
Job Monitor Options	274
Job Profiler	274
Navigation Techniques	275
Job Reports	275
Viewing reports:	276
Printing reports:	276
Refreshing reports data:	277
Job Forecast Report	277
Job Resource Consumption Report	277
Job Performance and Trends Reports	278
Job Timing and Conflicts Report	278
Database Load Balancing Report (Job Distribution)	279
Database Load Balancing Report (CPU Load)	279
Top 10 Jobs by CPU Usage	280
Top 10 Jobs by Memory Usage	280
Top 10 Jobs by Sorts	281
Top 10 Jobs by Disk I/O (Input/Output)	281
DB Application Expert	282
Overview	282
Application Monitor Console	283
Application Monitor Installation	284
Application Performance Statistics and Application Monitor Options	284
Application Performance Statistics	284
Application Monitor Options	285
Application Performance Reports	286
Application Resource Consumption Report	287
Application Performance and Trends Reports	288
Application Locks and Conflicts Report	288
Database Load Balancing Report	289
Top 10 Applications by CPU Usage	290
Top 10 Applications by Memory Usage	290
Top 10 Applications by Sorts	291
Top 10 Applications by Disk I/O (Input/Output)	291
Application Profiler Statistics (by Application)	292
Application Profiler Statistics (by User)	292
Application Profiler Statistics (by Terminal / Workstation)	292
Test Data Generator	293

Test Data Generator Overview	293
Working With Data Generator	293
Opening an Existing Project for Modifications	293
Creating a New Project.....	293
Deleting an Existing Project.....	295
Adding Tables to a Project.....	295
Removing Tables from a Project	295
Running Data Generation and Loading Process	296
Data Generation Methods and Options	296
Truncate Table	296
Disable Triggers	296
Number of rows.....	296
Load method	296
Considerations for choosing optimal loading method.....	297
Column-level Options.....	297
Data Libraries	299
How Random Are Generated Random Values?	302
DB Benchmark Expert.....	304
Overview	304
Running DB Benchmark Expert.....	304
Before You Begin.....	304
DB Zip/Unzip Expert	307
Overview	307
Zipping Tables.....	308
Unzipping Tables.....	309
Using DB Zip/Unzip for Quality Assurance and Development	310
DB Copy Expert	311
Overview	311
Copying Objects and Schemas	311
Getting started	311
Copying database tables	312
Copying other objects	314
Copying groups of objects	314
Copying entire schemas	315
Reports.....	315
Objects Report	315
Tables Report.....	315
Indexes Report.....	315
Clusters Report	316
Tablespaces Report	316
Database Files Report	316
Sequences Report	316
Snapshots Report	316
Synonyms Report.....	316
Database Links Report.....	316
Users Report.....	316
Table Privileges Report.....	316
Triggers Report	316
Procedures, Functions And Packages Report	316
Views Report.....	316
Parameters Report.....	317

Version Report	317
Custom Reports	317
Additional Tools	317
Analyzing object dependencies	317
Searching objects in the database.....	317
Finding invalid objects	317
Recompiling invalid objects	317
Recompile invalid objects using Database Object Navigator screen.....	317
Recompile invalid objects using Status Analyzer utility	318
Schema reverse-engineering.....	318
SQL Editor and Debugger.....	319
Overview	319
Creating new SQL script.....	319
Opening previously saved SQL script	319
Using drag-and drop.....	319
Importing text of an existing database stored procedure, function package, package body, type, type body or trigger	320
Comments	320
Variables	320
Edit commands, search and replace commands, go to commands	321
Supported SQL*Plus commands.....	321
Formatting commands.....	322
Entering SQL.....	322
Statement terminator.....	322
PL/SQL blocks support.....	323
Visual Explain Plan.....	323
Executing SQL	323
Tracing SQL and collecting performance statistics.....	324
Debugging SQL.....	325
Database Object Navigator	325
DBMS Output	329
Schema Reverse-engineering.....	329
SQL Navigator.....	330
PL/SQL Debugger	330
PL/SQL Debugger functions	331
Using the Debugger.....	331
Breakpoints.....	331
Debugger Views	332
Stepping Through Code.....	332
Evaluating/Modifying Variables.....	333
Stopping the Debugger.....	333
Hot Keys	333
Compiling Objects for Debugging	334
Version Specific Debugger Features	335
Using drag-and drop and quick schema browser	336
Comparing contents of SQL, TXT and other flat files	337
Customizing DBA Notepad.....	337
Table Relationships.....	338
Table Size Estimator	338
Database Object Finder.....	339
SQL Profiler.....	341

Overview	341
Setting up PL/SQL Profiler	342
Requirements.....	342
Working with PL/SQL Profiler Projects	343
To open a project.....	343
To start a new project	344
To modify a project	344
To delete a project.....	344
To start a new run	345
To delete a run from project.....	345
Profiling Applications and Analyzing Results	346
Analyzing results.....	346
Java Editor	347
Overview	347
Java Editor Interface	347
Java Editor Functions	348
Compiling and Loading Java Objects Into Database	348
Compiling Java class on the Oracle server from a Java a source code.....	348
Load a Java class already compiled on the client.....	348
File Loader.....	349
Overview	349
Loading Files Into Database.....	349
Loading files into existing tables	349
Loading files into new tables:.....	350
LOB Viewer and Loader	351
Overview	351
Open LOB Table for Browsing.....	351
Browsing and Modifying LOB Tables	352
Inserting a new record into a LOB table.....	353
Updating an existing record in a LOB table	354
Deleting an entire record from a LOB table	355
Deleting a document from a LOB table.....	355
Searching documents in a LOB table	355
Sorting and filtering documents in a LOB table.....	355
Printing documents in a LOB table	355
DB Tools Upgrade Assistant.....	356
Miscellaneous Information	357
Technical Support	357
Technical Support Request Form	357

About this guide

This manual describes the features of the DB Tools for Oracle product, including how to use the graphical user interface, available tools, functions and reports, how to install and configure optional back-end components.

Intended Audience

This document is for Oracle Database Administrators, Database Managers, and Database Developers.

Conventions used in this document

This section describes the style conventions used in this document.

Italic

An *italic* font is used for filenames, URLs, emphasized text, and the first usage of technical terms.

Monospace

A monospaced font is used for code fragments and data elements.

Bold

A **bold** font is used for important messages, names of options, names of controls and menu items, and keys.

User Input

Keys are rendered in **bold** to stand out from other text. Key combinations that are meant to be typed simultaneously are rendered with "+" sign between the keys, such as:

CTRL F

Keys that are meant to be typed in sequence will be separated with commas, for example:

Alt+S, H

This would mean that the user is expected to type the Alt and S keys simultaneously and then to type the H key.

Graphical symbols



- This symbol is used to indicate DBMS specific options and issues and to mark useful auditing tips.



- This symbol is used to indicate important notes.

Abbreviations and Product Reference Terms

DBMS – Database Management System

Oracle – This refers to all supported Oracle® database servers

The terms 'DB Tools for Oracle' and 'DB Tools' are used interchangeably in this document – they both refer to the same product]

Trademarks

DB Tools for Oracle, DB Audit, DB Audit Expert, DB Mail for Oracle, 24x7 Automation Suite, 24x7 Scheduler, SoftTree SQL Assistant are trademarks of SoftTree Technologies, Inc.

Windows 2000, Windows XP, Windows Vista are registered trademarks of Microsoft Corporation.

UNIX is the registered trademark of the X/Open Consortium.

Oracle is a registered trademark of Oracle Corporation.

All other trademarks appearing in this document are trademarks of their respective owners. All rights reserved.

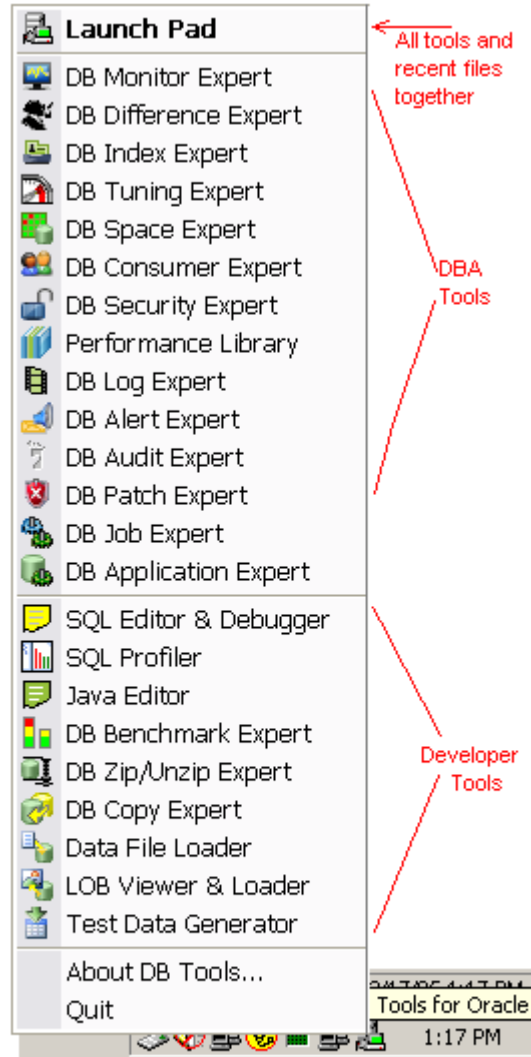
DB Tools Quick Start Menu

The Quick Start menu provides a quick and convenient way to access most DB Tools graphical components with just 2 mouse-clicks.

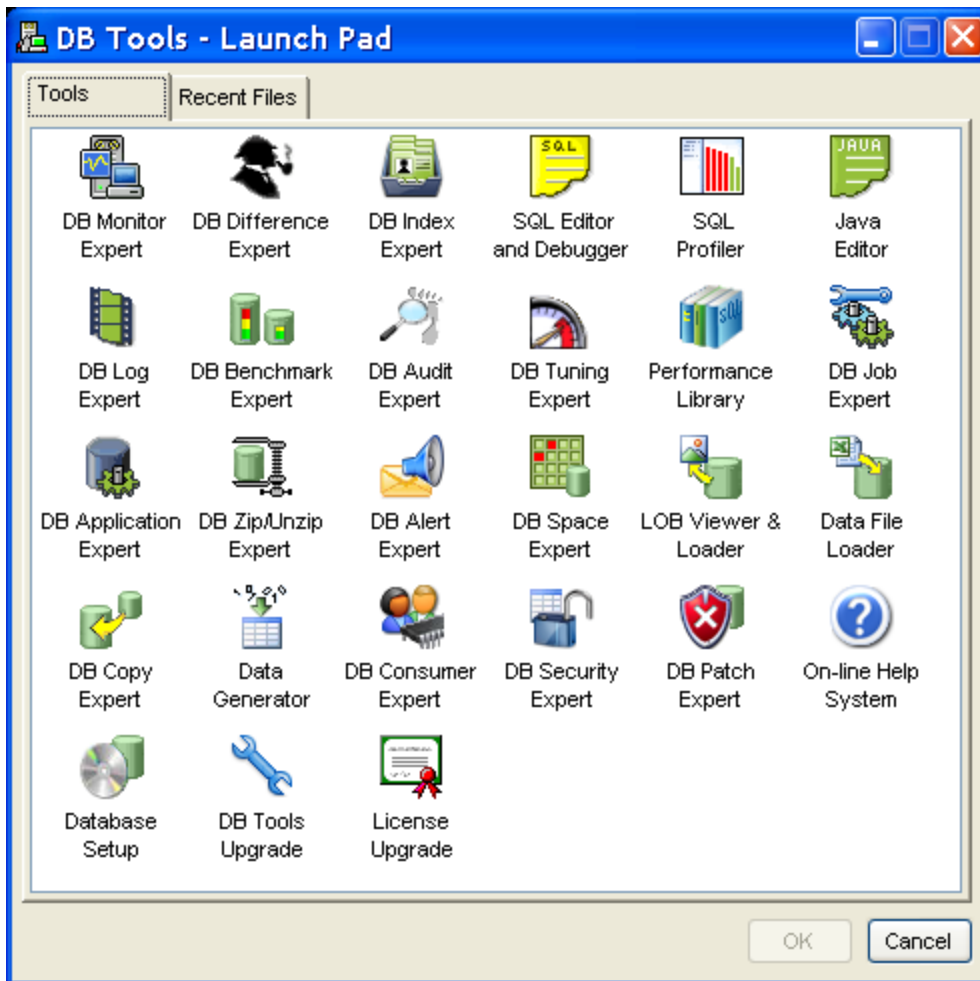
DB Tools Quick Start menu is normally started automatically on the computer startup and it runs as a small icon in the right-down corner of your screen.

To quickly access DB Tools components use any of the following methods:

- Right-click the icon and then selected in the DB Tools popup menu specific component you want to use.
- Double-click on the icon to open DB Tools Launch Pad. The Launch Pad screen provides direct access to all available components and also it provides quick access to 20 most recently opened SQL and Java files. To start a specific DB Tools component double on that component name or icon. To open a recent file switch to the files view and double-click on the file name. Double-clicking on a SQL file will open that file in DB Tools SQL Editor and Debugger. Double-clicking on a Java or JSP file will open that file in DB Tools Java Editor.



Right-click DB Tools icon




Common Dialogs and Controls

Application Toolbar

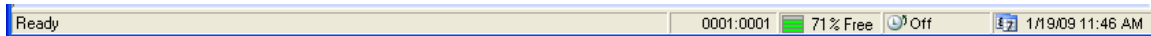


The toolbar displays buttons that duplicate commands in the menus. The buttons provide quick and easy access for commands.

Active toolbar items appear as 2D graphics until you move the mouse pointer over them. Disabled menu items have a silhouetted, engraved appearance and do not respond to the mouse pointer.

 **Tip:** To get help on the function of any toolbar button, rest the mouse pointer over the required button for a moment. DB Monitor Expert will display a tool-tip describing this button function.

Status Bar



The status bar displays the following information:

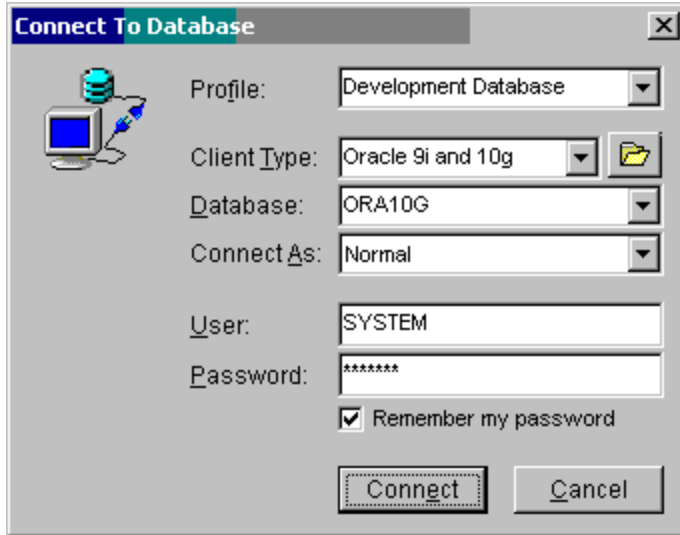
- Echo progress of work
- Current cursor position in the [DBA Notepad](#) in **row:column** format.
- Time left before first scheduled job run. "Off" indicates that there is when no jobs currently scheduled.
- Current date and time.

Connect to Database Dialog


Use the Connect dialog box to establish connections to your Oracle database servers. The same dialog box is used in all DB Tools components. The dialog can be displayed as a separate window or be embedded as page in a graphical multi-step component wizard.

Based on the information entered into this dialog, DB Tools automatically creates new and updates existing database profiles. The profile information is saved in the system registry. A database profile is a named set of connection parameters stored in the system registry under DB Tools key. On consecutive runs of any DB tools component you can pull the required profile name from the Profile drop-down list and then click the Connect button to establish a connection to your database server.

Profile – Descriptive database connection name. Enter a descriptive name for a database you connect to, for example Data Warehouse, Dept. DB, etc. The profile name can contain any text up to 50 characters long, including spaces. If the profile already exists, select it from the Profile drop-down list: DB Tools will automatically populate the rest of the connection dialog fields.



Client Type– Type of the Oracle database client software installed on your computer. DB Tools uses Oracle client software to communicate with your Oracle database servers. For more information on this process read [How DB Tools Talks to Oracle](#) topic.

 **Important Notes: Do not confuse Oracle client software version installed on your computer with the Oracle database server version you are connecting to. Always choose the proper client version in the Client Type drop-down box.**


In case if you have multiple Oracle clients installed and don't want DB Tools to guess which one of them to use for the

connection use the **Browse**  button

to locate the correct client. You must select the BIN directory containing Oracle client files. Note that if you use the Browse function and correctly locate your Oracle client BIN DB Tools will automatically determines your Oracle client version and select the correct value in the Client Type drop-down list.

Depending on the selected client type DB Tools internally uses different database drivers for different Oracle client versions.

For ODBC connections select ODBC value for the client Type. Please note that all ODBC drivers use the same Oracle client software used by DB Tools native database drivers. For more information on how to configure ODBC connections see [Connecting to Oracle using Oracle ODBC driver](#) topic.

 **Important Notes:** ODBC connections are currently supported for backward compatibility with previous DB Tools versions. ODBC connections are usually unreliable and should be avoided whenever possible. It is highly recommended that you always use native database

Database – Choose the desired database server name from the server drop-down box. If you cannot find your server name in the list, type it in the edit field. You should enter the same name that you enter in SQL*Plus and in other Oracle tools. This name is often called *Oracle TNS Name*. You can optionally enter the complete TNS string instead of the name. In the latest case the string must be entered as a single line.

Connect As – This parameter controls database connection mode. In most situations you should use *Normal* connection mode. When connecting to Oracle 9i and up databases using SYS user account always pick *SYSDBA* connection mode. Oracle will not allow you to connect as SYS in the *Normal* mode.


 **Important Notes:**

- SYSDBA and SYSOPER connections are only available with the Oracle 9i and 10g client software. You must have Oracle 9i or better client installed on your system in order to use this option.
- Because of the Oracle restrictions you cannot use Oracle client 10g or better to connect to old Oracle database versions 8i and prior. To connect to and work with these databases you must have an older version of the Oracle client software installed on your system. Fortunately to connect to Oracle 8i database or prior version as SYS user you don't need to use the SYSDBA option. If you wish you may have multiple Oracle clients installed on your computer. As mentioned earlier in such situations use the **Browse** button to select the directory containing Oracle client software that you want to use for the connection.

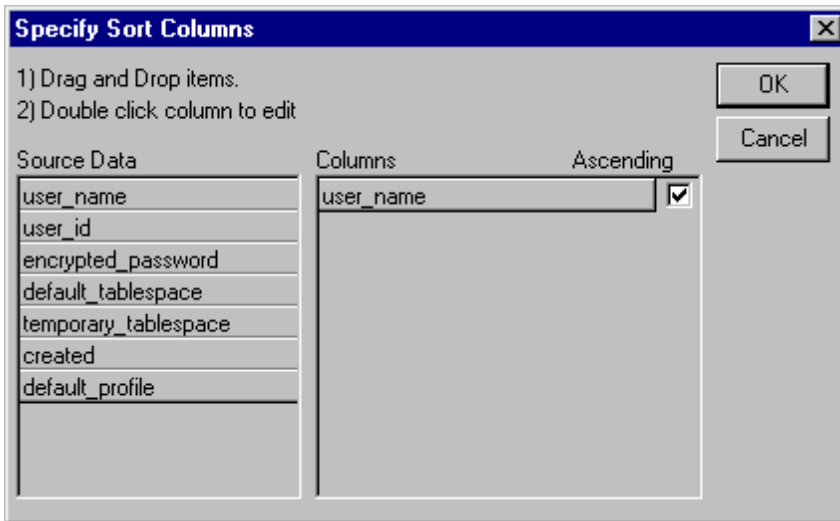
User – For non-trusted connections, provide the database user name for the server you are connecting to. For trusted connections, do not enter anything in this field.

Password – For non-trusted connections, provide the password for the user or login name. For trusted connections, do not enter anything in this field.

Remember my password – Use this option to allow DB Tools to remember your database password so that next time you connect to the database you don't have to type it in again. If this option is checked, the password is stored in the local system registry along with other profile information.

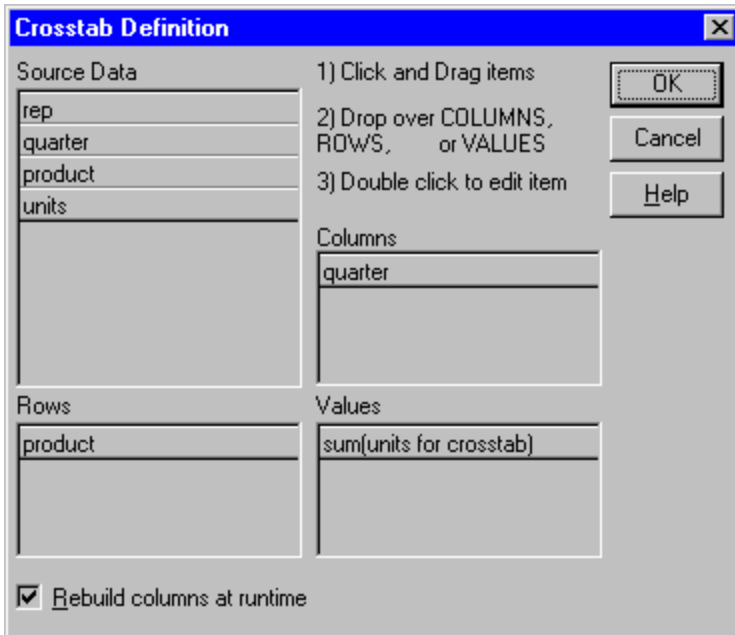
 **Note:** This information is only as secure as the secured access to your computer. If you share this computer with other non-privileged users, do not use this option as it compromises your system security.

Sort Dialog



For advanced sort, double-click the column from column list. The [modify expression dialog box](#) will appear. See [functions](#) topic for help on functions syntax and usage.

Crosstab Definition Dialog



Updating Crosstab Definition

1. Use drag and drop to fill in the boxes for Columns, Rows, and Values.
2. Click OK button. DB Tools changes the definition of the crosstab.



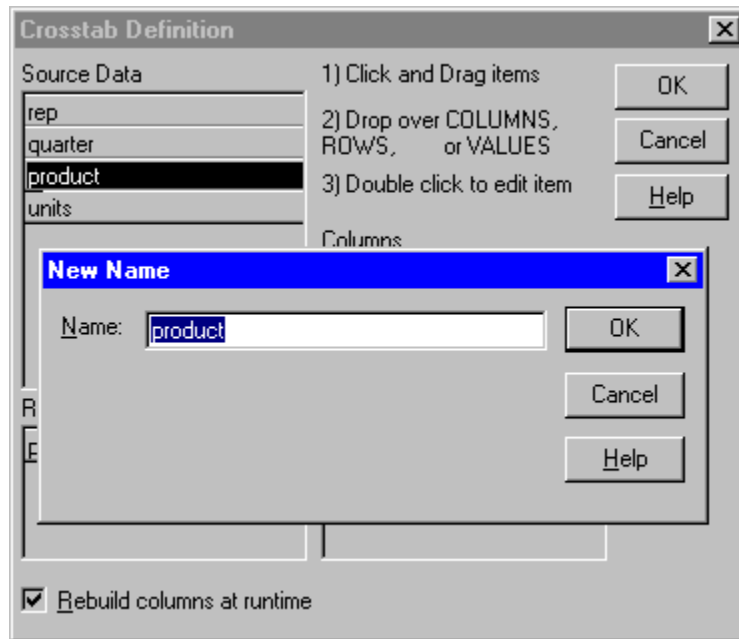
Tip: When you define a crosstab with more than one column, the order of the columns in the Columns box of the Crosstab Definition dialog box governs the way DB Tools group the columns. To let DB Tools define the most effective expressions, make the column that contains the grouping values (for example, year) the first column in the Columns box and the column that contains the values to be grouped (for example, bytes) second.

Changing the names used for the columns and rows

Sometimes names of columns in the database are not very user-friendly and may not be meaningful. You can change the names that are used to label rows and columns in Crosstab so you better understand the data.

To change the names used in crosstabs:

1. Double-click the name of the column in the Source Data box in the Crosstab definition dialog box. The New Name dialog box displays.



2. Specify the name you want to be used to label the corresponding column. You can have multiple-word labels by using underscores: underscores are replaced by spaces in the Preview tab and during execution.
3. Click OK button. DB Tools changes the column name in the Source Data box and anywhere else the column is used.

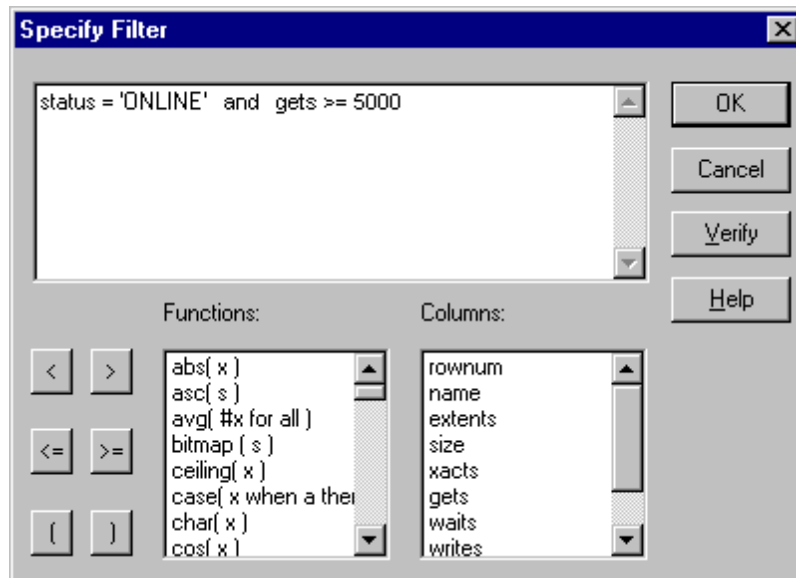


Notes:

- When DB Tools generates a Crosstab, the columns and rows are automatically totaled for you.
- There are five special functions you can use only in crosstabs: [CrosstabAvg](#), [CrosstabCount](#), [CrosstabMax](#), [CrosstabMin](#), and [CrosstabSum](#).

For advanced definition, double-click the column from the Columns, Values or Rows lists. The [modify expression dialog box](#) will appear. See [functions](#) topic for help on functions syntax and usage.

Filter Dialog



To specify a filter:

1. Define the filter by entering text in the filter definition box and inserting the [operators](#), symbols, [functions](#) and column numbers where appropriate.



Tip: You can use the following syntax to reference the original value of a column in any expression:

column.original - where column is the name of the column.

You can use column numbers instead of column names. Prefix column number with # symbol:

#2 - substitutes `department_head_id` column which goes second in select statement as `department.head_id`

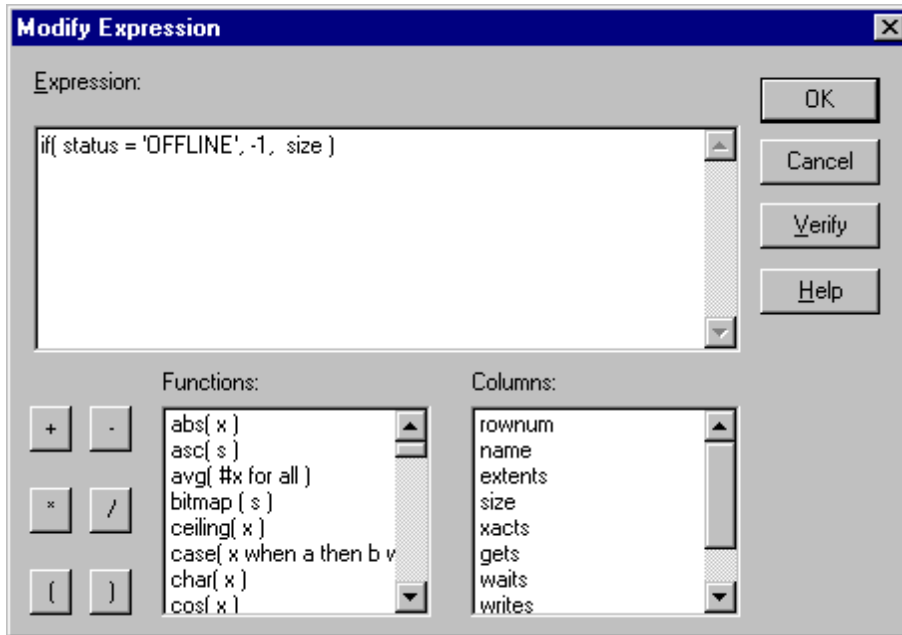
2. Repeat the procedure until the complete filter expression is defined.




Tip: To verify the expression without saving it, click the Verify button. A message displays, telling you whether the expression is valid.

3. Click OK. The syntax is verified. If there is an error in the syntax, a message displays. If no errors are found, the Filter Definition dialog box closes and DB Tools applies the filter.

Modify Expression Dialog



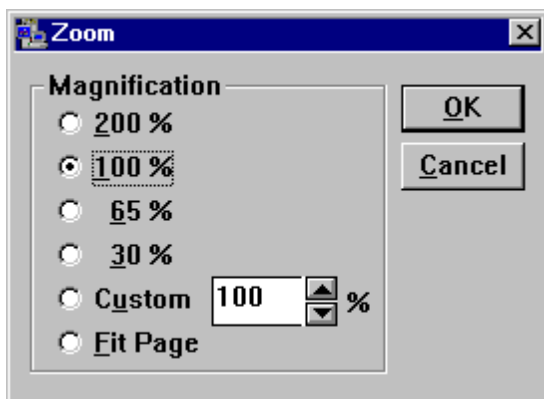
Build an expression using [operators](#) and embedded [functions](#). You can paste column names and functions from the Functions and Columns lists. To refer to the value of a column in a specific row: Enter: `columnname [integer]` where `columnname` is the name of the column and integer indicates the current row.

 **Tip:** A negative integer indicates a previous row. An integer preceded by a plus sign (+) indicates a successive row.

To verify the expression without closing the dialog box:
Click Verify. A message box displays indicating whether the expression is valid.

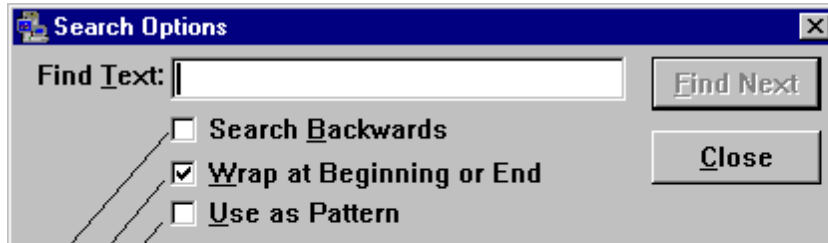
Example: `if (status = 'INVALID', -1, id)`

Zoom Dialog



See also:[Working with data and export](#)

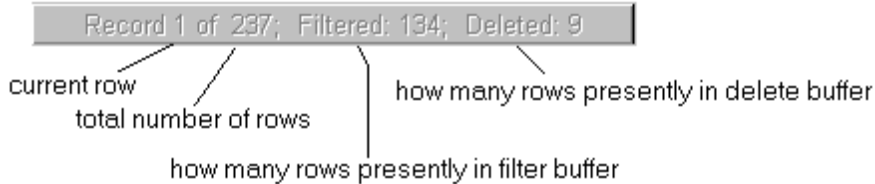
Search Dialog



- Use text entered in Find Text field as a search pattern
- After search performed from current row to the last one continue search from first row
- Search backwards (from current row to the first one)

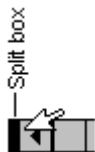
See also:[Working with data and export](#)

Row Descriptor Control

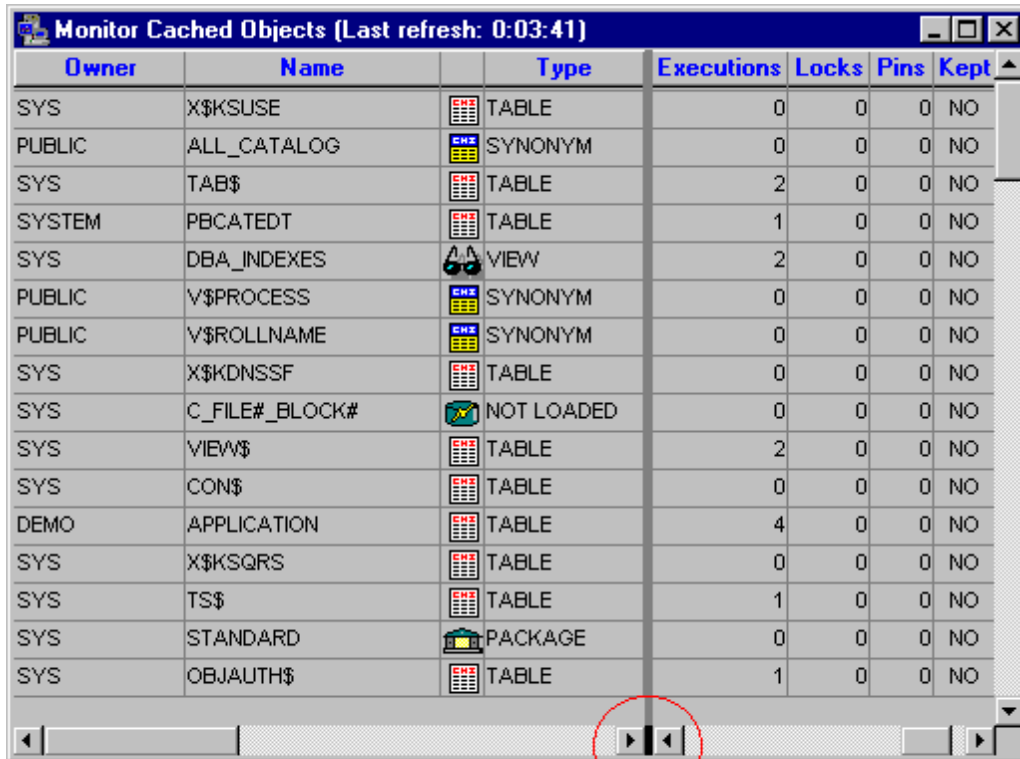
**See also:**Navigation section from [Working with data and export](#) topic

Split Control

For your convenience DB Tools provides [split control](#) that you use to view two parts of a datasheet or report simultaneously. The split handle is displayed in the lower left corner between the left edge and the horizontal scroll bar.



To split the visible area into two panes with separate scrollbars, place mouse pointer over the split handle and drag it to the right.



Owner	Name	Type	Executions	Locks	Pins	Kept
SYS	X\$KSUSE	TABLE	0	0	0	NO
PUBLIC	ALL_CATALOG	SYNONYM	0	0	0	NO
SYS	TAB\$	TABLE	2	0	0	NO
SYSTEM	PBCATEDT	TABLE	1	0	0	NO
SYS	DBA_INDEXES	VIEW	2	0	0	NO
PUBLIC	V\$PROCESS	SYNONYM	0	0	0	NO
PUBLIC	V\$ROLLNAME	SYNONYM	0	0	0	NO
SYS	X\$KDSSSF	TABLE	0	0	0	NO
SYS	C_FILE#_BLOCK#	NOT LOADED	0	0	0	NO
SYS	VIEW\$	TABLE	2	0	0	NO
SYS	CON\$	TABLE	0	0	0	NO
DEMO	APPLICATION	TABLE	4	0	0	NO
SYS	X\$KSQRS	TABLE	0	0	0	NO
SYS	TS\$	TABLE	1	0	0	NO
SYS	STANDARD	PACKAGE	0	0	0	NO
SYS	OBJAUTH\$	TABLE	1	0	0	NO


activated split control

See also:

Navigation section from [Working with data and export](#) topic

Sampling Cycle

All monitors and real-time graphs by default perform automatic sampling. To trigger sampling manually, select **File/New Refresh** command (shortcut: CTRL N). To change sampling frequency (refresh rate), do the following:

- 1 Select **Tools/Options** command or simply click the Options button on the [toolbar](#) . An [Option dialog](#) box will appear.
- 2 Click on the **Timer** tab page.
- 3 Select required monitor name from the **Screen** drop-down list.
- 4 Enter appropriate number of seconds in the **Interval** field. Note that you can set timer interval from 1 second to 2.7 hour (9999 seconds)
- 5 Click the **OK** button



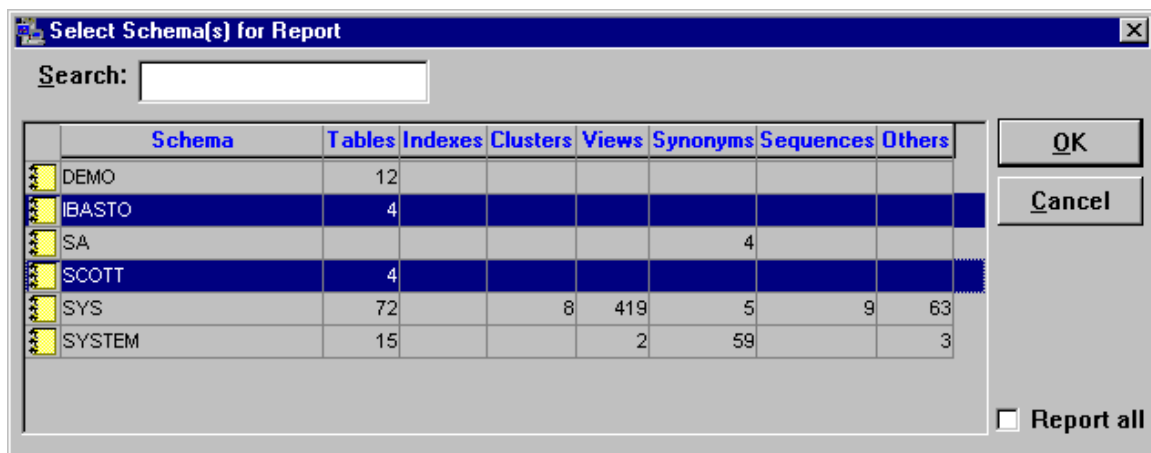
Tip: To completely disable automatic sampling, enter **0** in the **Interval** field.

Schema Names Cache

For improving overall system performance DB Tools by default caches database schema names. This means that DB Tools retrieves schema names only once. There are two kinds of schema caches. One is for all database schemas and another is for schemas that own database tables. Schema name caching is optional. You can turn on/off this feature in the program [Options dialog](#).

Schema Selection Dialog

Schema selection dialog provides you with flexibility necessary when dealing with large Oracle databases. You can select one or more schemas for which DB Monitor Expert will generate the chosen report or run the requested analysis or search operation. In addition, schema selection dialog displays summary statistics for each schema.



To select multiple schemas

Uncheck **Report All** checkbox. Hold down the CTRL key, and then click each schema you want to select. To select range of schema, hold down the SHIFT key and then click on first and last schema from the range. To select all schemas use shortcut CTRL +, to deselect all schemas use shortcut CTRL -.

Note: In order to select all schemas in the database you either selection of all schemas from the list or check **Report All** checkbox. However maximum number of selected schemas from the list may not exceed 255, opposite to unlimited number when you check **Report All** checkbox.

To find a schema:

To quickly find the required schema, type schema name in search box. DB Monitor Expert will scroll to first match that starts with typed text.

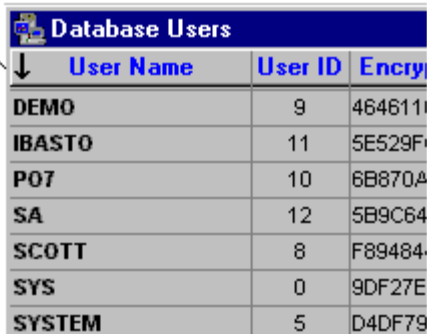
Working with data and import/export

Sorting

You can sort the rows of your datasheet by the data in one or more adjacent columns.

For simple sort by one column, click the header of the field you want to sort by. The arrow down appears over the header indicating data sorted in ascending order. Click the header again to sort descending. The arrow up appears over the header indicating data in descendant order.

Arrow down indicates ascendant sort



↓ User Name	User ID	Encry
DEMO	9	464611
IBASTO	11	5E529F
P07	10	6B870A
SA	12	5B9C64
SCOTT	8	F89484
SYS	0	9DF27E
SYSTEM	5	D4DF79

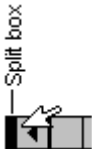
For complex sorts by two or more columns or even by expressions, click View/Sort menu. The [sort dialog box](#) will appear. Drag and drop columns from the source data list box to the columns list box. Check the ascending column (checkbox) next to the dragged column/expression to sort ascending or leave it blank to sort descending. For advanced sort, double-click the column from column list. The [modify expression dialog box](#) will appear. See [functions](#) topic for help on functions syntax and usage.

Filtering

You can limit the data you see in your datasheet by filtering it. To apply filter, click View/Filter menu. The [specify filter dialog box](#) will appear. That dialog box is similar to the [modify expression dialog box](#). See [modify expression dialog box](#) for valid expression examples. See [functions](#) topic for help on functions syntax and usage.

Navigating

To navigate through data in a datasheet you can use standard keyboard navigation keys. For quick scrolling to the required row, you can use either [navigation buttons](#) on the bottom side or click **Edit/Go To Row** command in the menu.



For your convenience DB Tools provides [split control](#) that you use to view two parts of a datasheet simultaneously. The split handle is displayed in the lower left corner between datasheet's left edge and the horizontal scroll bar. To split the visible area into two panes with separate scrollbars, place mouse pointer over the split handle and drag it to the right.

The current row always has arrow symbol displayed over the row header. .

To move between fields use TAB key (this is applicable only if the datasheet is modifiable).

Interrupting data retrieval

To interrupt the long running data-retrieval, use **View/Stop Retrieving** command (shortcut F9) or click the Stop button on the application toolbar. .

Printing

Any DB Tools screen can be printed. Use **File/Print** command or shortcut CTRL P. You can preview how it will look when printed by choosing **File/Print Preview**.

See: [Print](#) topic for information on print and print preview options.

Selecting text in a modifiable field

- 1 Selecting text in a modifiable field by using the mouse
 - Any amount of text - Drag over the text
 - A word - Double-click the word
- 2 Selecting text in a modifiable field by using shortcut keys
 - Select text by holding down SHIFT and pressing the key that moves the insertion point.
 -

To extend a selection	Press
One character to the right	SHIFT+RIGHT ARROW
One character to the left	SHIFT+LEFT ARROW
To the end of a word	CTRL SHIFT+RIGHT ARROW
To the beginning of a word	CTRL SHIFT+LEFT ARROW
To the end of a line	SHIFT+END
To the beginning of a line	SHIFT+HOME
One line down	SHIFT+DOWN ARROW
One line up	SHIFT+UP ARROW

Selecting fields, columns, rows in a datasheet

Selected fields, columns and rows have black background and white text color. Data in selected areas can be copied to the clipboard for sharing with other Windows programs.

Click on the column header while holding down the CTRL key to select the entire column. To select a row, click on the row header. To select range of rows, first select first row, then hold down SHIFT key and click on the last row in the range. You may have any combination of selected columns, rows and individual fields



Tip: To preserve existing selection and select additional rows, columns or fields, always remember to hold down CTRL key before you click on column, field or row header.

To select all rows click **Edit** menu then **Select all** command.

Create Date	House No	Street	Name	Apartment
4/25/86 00:00:00	207	EAST 61ST STREET		
4/25/86 00:00:00	1035	FIFTH AVENUE		
4/25/86 00:00:00		6TH AVENUE	Waverly Place	WAV
5/9/86 00:00:00	179	WILSON AVENUE	3rd Police Prec.	
5/7/86 00:00:00	2360	ADAM CLAYTON POWELL, JR. E	201 west 138th street	
5/8/86 00:00:00	328	WEST 4TH STREET	AKA 38 8th Street	
4/25/86 00:00:00	170	WAVERLY PLACE	aka 98 gravezst	
4/25/86 00:00:00			American Mus. of natural history	
5/8/86 00:00:00	800	9TH STREET	Church of the Apostles	
4/16/87 00:00:00	712	FIFTH AVENUE	caty rizzali	
5/7/86 00:00:00	641	WASHINGTON STREET	Federal Archive BLDG	
5/8/86 00:00:00	35	EAST 76TH STREET	Hotel Carlyle 17th & 18th floors	
5/8/86 00:00:00	35	EAST 76TH STREET	Hotel Carlyle 8th floor	
5/8/86 00:00:00	35	EAST 76TH STREET	Hotel Carlyle 8th floor	

Record 229 of 249

Example datasheet with selected data

Copying

Use **Edit/Copy** (shortcut: CTRL C) command to copy selected data to the Windows Clipboard. Copied data have tab-separated columns with a return at the end of each row.

Editing

You can edit data in your result datasheets and save them back to the database. However, not every datasheet can be modifiable. All modifiable datasheets as well as all input fields have white background. All not modifiable datasheets and input fields have gray background. When DB Tools executes your SELECT query from the [DBA notepad window and executing SQL commands](#) or retrieves data from the selected table (see [Quick data view and data manipulation](#)), it checks the following conditions:

- The result set comes from a single table (Yes/No)
- The table has primary key or unique constraint (Yes/No)

When both conditions evaluate to **Yes** DB Tools knows how to update the table if any changes will be made. In that case, the result datasheet can be modified, otherwise DB Tools prevents making data changes in the datasheet.

In non-modifiable datasheet, symbol  is displayed in the upper left corner.

You can use all standard Windows shortcuts for working with data, such as copy (CTRL C), paste (CTRL V), cut (CTRL X), undo (CTRL Z), and redo (CTRL Y). DB Tools supports multi-level Undo/Redo operations.

Use the following techniques to manipulate rows of data:

Action	Result
Select File/New Row command (shortcut: CTRL N)	Insert a new row before current row
Select File/Delete Row command (shortcut: CTRL D)	Delete current row
Right mouse click on any row causes context menu appears. Select Insert Row from the menu	Insert a new row before clicked row
Right mouse click on any row causes context menu appears. Select Delete Row from the menu	Delete clicked row

For your convenience DB Tools displays row modification status over the row header. The following table describes supported row statuses:




Picture	Row Status	Save Action
	Newly inserted blank row	[None]
	Newly inserted and modified row	INSERT
	Retrieved and modified row	UPDATE

TABLE: DEMO.APPLICATION (Response: 00:00:00:132, Retrieve: 00:00:03:077)					
Docket	Create Date	House No	Street	Name	
861011	4/23/86 00:00:00	132	LAFAYETTE AVENUE		
860890	4/25/86 00:00:00	1220-1227	FIFTH AVENUE	museum of the city of NY	
860843	4/25/86 00:00:00	967	MADISON AVENUE		
860812	4/25/86 00:00:00	925	PRESIDENT STREET		
860909	5/5/86 00:00:00	408	HENRY STREET		
860912	5/5/86 00:00:00	148-150	WAVERLY AVENUE		
860709	5/8/86 00:00:00	328	WEST 4TH STREET	AKA 38 8th Street	
860986	1/1/80 00:00:00	128	STATE STREET		
861006	4/23/86 00:00:00	117	MILTON STREET		
860795	5/9/86 00:00:00	545	BROADWAY		
860904	5/5/86 00:00:00	47-49	GREENWICH AVENUE		
860745	5/9/86 00:00:00	809	MADISON AVENUE		
860866	4/25/86 00:00:00	24	TOMPKINS PLACE		
860670	5/7/86 00:00:00	82-86	BEDFORD STREET		

Record 34 of 371; Deleted: 9

Example of a changed datasheet

Saving

Select **File/Save** (shortcut: CTRL S) to save made changes. If a database error occurs during the save operation, DB Tools automatically executes the ROLLBACK command and displays the error message, explaining the problem. If the save operations succeeds then the database COMMIT is automatically executed.

Exporting data

You can export data from your result datasheets to a number of popular file formats (see table below). You also can copy part or all of the contents of any datasheet (all reports are regular datasheets) and paste it into another Windows application by way of the Windows Clipboard function. See previous topics [Selecting fields, columns, rows in a datasheet](#) and [Copying data](#).

File Extension	File Format
CSV	Comma-separated values
DBF	DBASE-II format
DBF	DBASE-III format
DIF	Data Interchange format
XLS	Microsoft Excel format
HTM	Text with HTML formatting (HTML table)
SQL	SQL syntax (insert statements)
SLK	Microsoft Multiplan format
TXT	Tab-separated columns with a return at the end of each row
WKS	Lotus 1-2-3 format
WK1	Lotus 1-2-3 format
WMF	Windows Metafile format

To Export to a File:

1. Select **File/Export** (shortcut: ALT+F, R) from the menu. The standard Save dialog opens:

2. Specify the file name and choose file format by selecting one from the Save File as Type drop down list.
3. Click the **OK** button to save the result set to the specified file format.

See also:


[DBA notepad window and executing SQL commands](#)
[Quick data view and data manipulation](#)


Importing data

You can import data to your result datasheets from a number of popular file formats (see table below).

File Extension	File Format
CSV	Comma-separated values
DBF	DBASE-II format
DBF	DBASE-III format
TXT	Tab-separated columns with a return at the end of each row

To Import from a File:

1. Select **File/Import** (shortcut: ALT+F, I) from the menu. The standard Open File dialog opens:
2. Specify the file name and choose file format by selecting one from the Open File as Type drop down list.
3. Click the **OK** button to import data to the active datasheet.
4. Select **File/Save** (shortcut: CTRL S) from the menu or simply click the Save  button to save imported rows to the database.


 **Note:** Number of columns in the import file must equal or less than number of columns in the datasheet. You cannot save imported data in the database when the datasheet is not updateable, in other words when the SELECT statement for the datasheet does not include all primary key columns.

See also:

[Exporting Data](#)
[DBA notepad window and executing SQL commands](#)
[Quick data view and data manipulation](#)

Searching

Full search:

1. Select **Edit/Find Text** (shortcut: CTRL F) command from the menu or click the Find button on the application toolbar.  A [search dialog box](#) will appear.
2. Type in the text you want to search for.
3. Choose the required [search options](#)
4. Click the **Find Next** button to start searching. Note that the search is performed in all columns of the active datasheet. The search is not case sensitive.
5. If specified text is found, the search dialog box continues staying on top so you can review the found text and continue searching if needed. The found text is highlighted on the report and the report is automatically advanced to the page where the text is found. You can keep clicking the Find Next button to continue searching for the next occurrences of the specified text, otherwise click the Close button to stop search.

See [metacharacters and sample patterns](#) topic for help on valid search patterns.

Quick search: this search is performed only in one column.

1. To run a quick search, right-click on the column in which you want to find something. A popup menu will appear.
2. Select the **Quick Search** command. The Quick Search dialog box will appear.
3. Type in the text you want to search for.
4. Click the **OK** button. Note that search will be performed in the clicked column only starting forward from the clicked row.

Zooming in/out

You can "zoom in" to get a close-up view of your data or "zoom out" to see more of the page at a reduced size. Click **View/Zoom** menu. A [zoom dialog](#) box will appear. Select appropriate magnification and click **OK** button

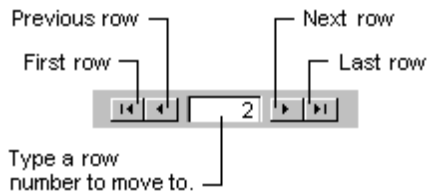
Resizing columns and rows

To change column width, place mouse pointer over edge of column header and drag it to obtain required width.

You can use the mouse to change the height of the rows. Place mouse pointer over row grid line and drag it to obtain required height.

Navigation Buttons

To quickly move between rows, you can use the navigation buttons.



Customizing DB Tools Interface

The application can be reconfigured to some extent. You can customize the following features:

- Application toolbar. For help on this see [application toolbar](#) topic.
- Timer interval for real-time monitors and graph as well as you can customize number of checkpoints for real-time graphs (see **timer** section bellow).
- How DB Tools performs various database operations (see **database operations** section bellow).
- [DBA Notepad](#) font (see **editor** section bellow).
- Printing (see **print** section bellow).
- Debugging

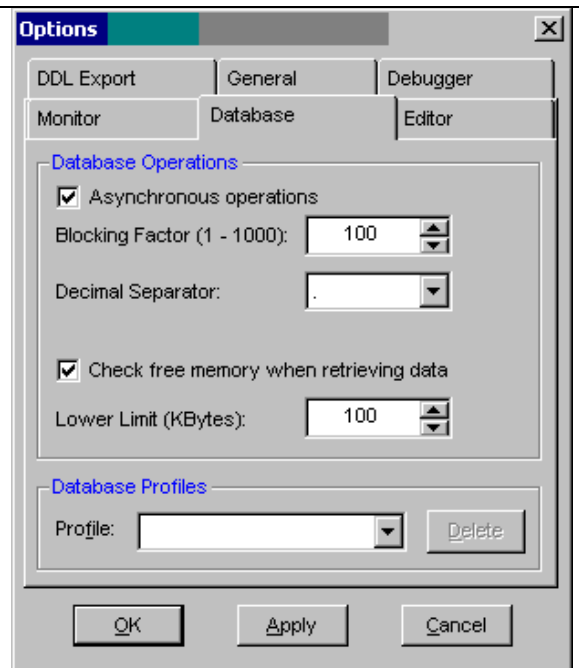
Timer Options

1. Click the **Timer** tab.
2. From the **Screen** drop-down list box select one of the available real-time monitors or graphs you want to customize.
3. Specify required refresh rate in the **Timer Interval** field. Note that **0** will stop the timer of the selected screen.
4. Specify required number of checkpoint in the Checkpoints field. Note that the **Checkpoints** field is available only for real-time performance graphs.
5. Click the **Apply** button to save settings.
6. If you want to customize another screen, repeat step 1 to 4.



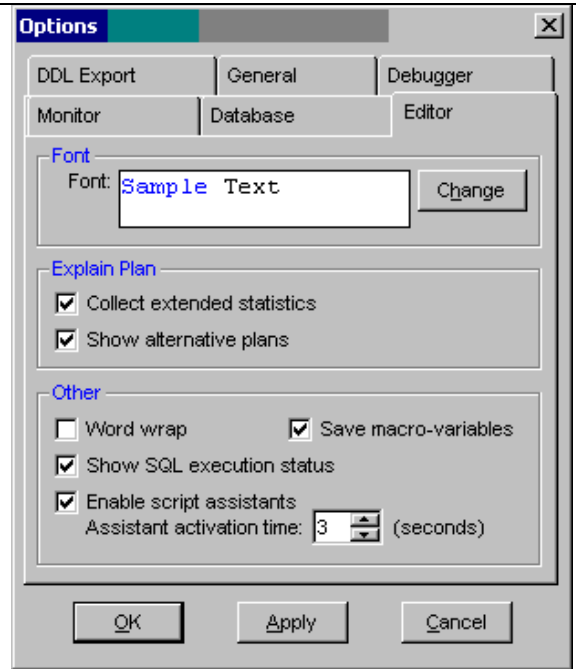
Database Operations Options

1. Select the **Database** tab.
2. Check asynchronous operation box to allow DB Monitor Expert perform asynchronous operations on your database. Note that not all environments support asynchronous operations.
3. Correct value in the **blocking factor** field. The blocking factor determines the number of rows that DB Monitor Expert report or result datasheet can fetch from the database at one time. Choosing right block factor can improve performance when accessing database.
4. Choose the **check memory on retrieve** option to enable memory check. If the check is enabled then DB Monitor Expert will check free memory after each 10 retrieved rows. This process slows performance, but ensures that there will not be insufficient memory problem.
5. Enter appropriate for your system value in the **Lower Limit** field. DB Monitor Expert interrupts data retrieval when the memory check option is enabled and amount of free memory is less then specified in the **Lower Limit** field.
6. Click the **Apply** button to save settings.



Editor Options

1. Select the **Editor** tab.
2. To change font in the DBA Notepad click the **Change** button. A standard Windows font dialog box will appear.
3. Choose required font.
4. Click the **OK** to close font dialog.
5. Click the **OK** button to save settings.

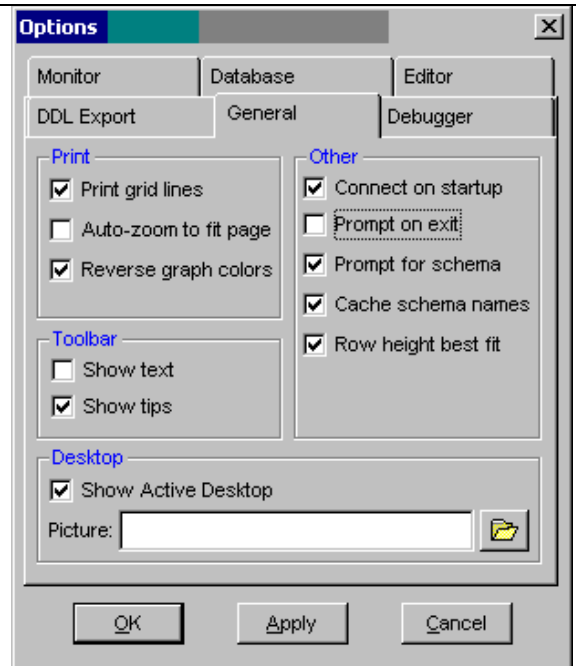


Print Options

You may disable output for grid lines when printing DB Monitor Expert reports, graphs, and datasheets.

1. Click the **General** tab.
2. Uncheck the **print grid lines** box to disable printing of grid lines.
3. Check the **auto-zoom** box to allow DB Tools automatically zoom out the documents that do not fit across the page. Note that this affects only print output.
4. By default, most graphs have black background for improved on screen viewing. Check the **reverse graph colors** box if you want Tools print graphs on white background.
5. Click the **OK** button to save settings.

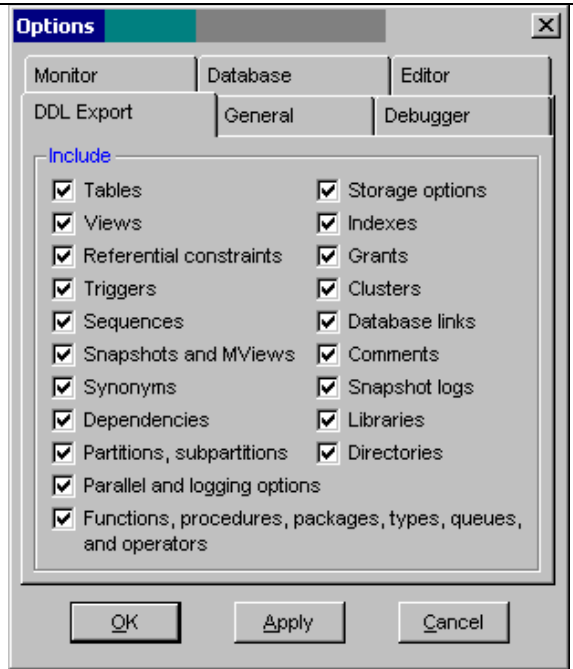
See: [Print](#) topic for information on other print and print preview options.



DDL Export Options

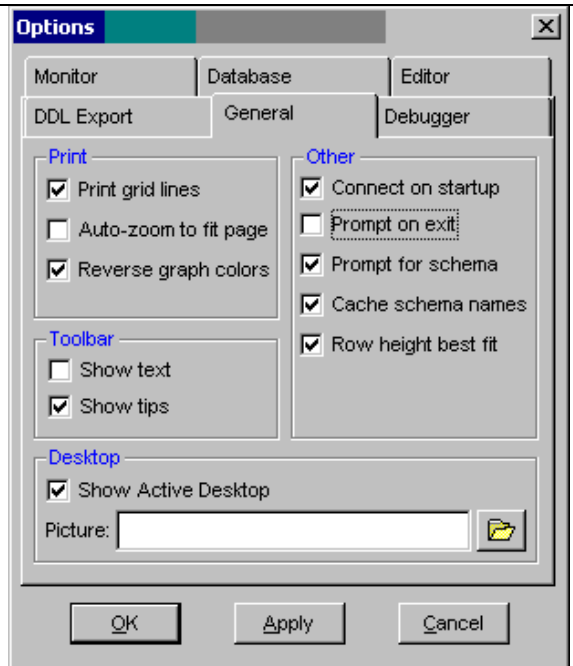
On this tab, you can set defaults for DDL (data definition language) generated by Tools when extracting schema, table, and view syntax.

1. Click the DDL Export tab.
2. Check required DDL options you want to make default.
3. Click the OK button to save settings.



General Options


1. Click the General tab
2. Check the prompt on exit box if you want DB Monitor Expert asks confirmation for quitting.
3. Check the prompt for schema box if you want DB Monitor Expert to display [Select Schema dialog](#) for reports and tools.
4. Click the OK button to save settings.



Printing

Choose right printer

Select **File/Print Setup** command. A Printer Setup dialog box will appear. From printer list, select a printer and click **OK** button.

 **Note:** Most print setup options are accessible in print preview mode only (see next section). When you use **File/Print** (shortcut: CTRL P) command print defaults are used. Each time when you print from the print preview mode, print options dialog box appear. You can choose print options for the document, which is being printed.

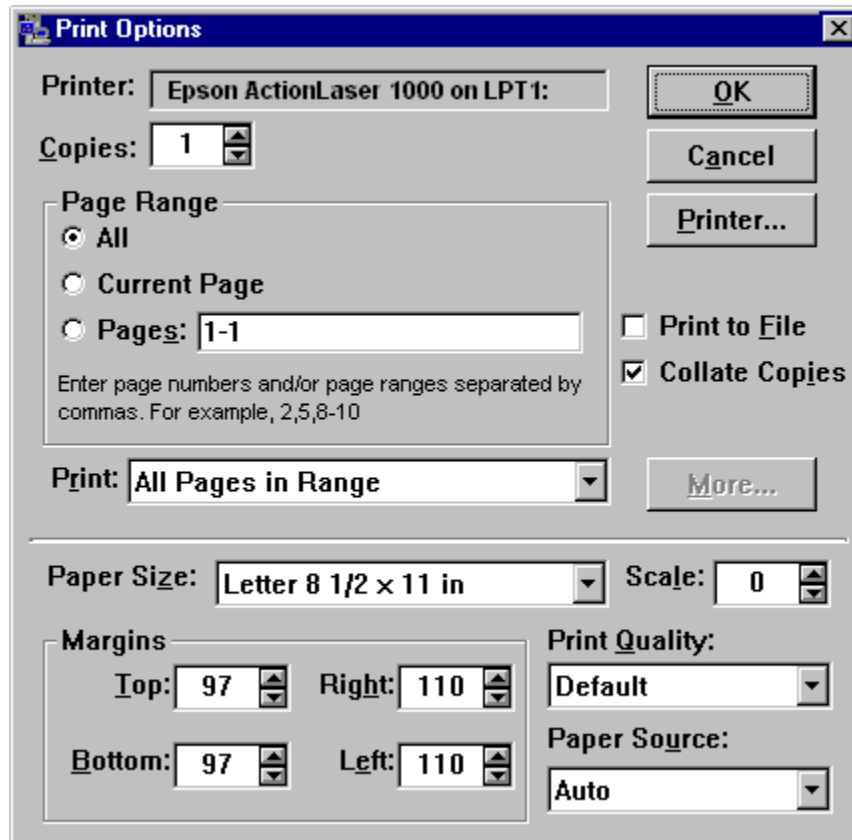
Preview a document before printing

To display each page as it looks when printed, choose Print Preview.

Prevent a document from flowing onto an additional printed page

If the screen page is too wide to fit on one printed page and only a small amount of text appears on the next page, you may narrow column width for some columns. To narrow column put mouse pointer over right edge of the column header and drag it to the left until it reaches required width. If the screen page is much wider than printed page you can use **Data/Zoom** command to "zoom out" the document so you will see more of the document at a reduced size on each page.

Print Options



Print Options

Printer: Epson ActionLaser 1000 on LPT1: OK

Copies: 1 Cancel

Page Range

All Printer...

Current Page

Pages: 1-1 Print to File

Enter page numbers and/or page ranges separated by commas. For example, 2,5,8-10 Collate Copies

Print: All Pages in Range More...

Paper Size: Letter 8 1/2 x 11 in Scale: 0

Margins

Top: 97 Right: 110

Bottom: 97 Left: 110

Print Quality: Default

Paper Source: Auto

Print Options Dialog

Print a range of pages or print specific pages only

Under **Page Range**, specify the portion of the document you want to print. Enter the page numbers or page ranges you want to include, or both. You can separate by commas individual pages and page ranges. For example, to print pages 1, 2, and 5 through 10, enter: "1,2, 5-10". The empty string means print all. From the **Range Include** drop-down list box choose what pages to print within the required range. Values are: print all, print all even pages; print all odd pages.

Print more than one copy at a time


- **Collate** - Whether printing is collated. Note that collating is usually slower since the print is repeated to produce collated sets.
- **Copies** - The number of copies to be printed.

Print To File

When you print to a file, a standard select file name dialog box appears. Enter name of file or click Cancel button to cancel printing. Later you can copy file to the required printer port.

Print Margins

- **Bottom** - Width of the bottom margin on the printed page.
- **Left** - Width of the left margin on the printed page.
- **Right** - Width of the right margin on the printed page.
- **Top** - Width of the top margin on the printed page.

 **Note:** When you are in the print preview mode, you can use drag-and-drop to visually adjust page margins. You must have rulers visible (keep the "ruler visible" option tuned on). Place mouse pointer over resize arrows displayed on the rulers. Drag it to adjust margins as needed.

Print Preview - Margin Marks

Other Print Options

- **Orientation** - Print orientation.
- **Paper Size** - A size of the paper that will be used for the output.
- **Paper Source** - The printer's bin that will be used as the paper source.
- **Print Quality** - Indicating the quality of the output. Note that higher quality is usually slower.
- **Scale** - An integer specifying the scale of the printed output as a percent. The scaling percentage is passed to the print driver. If you have problems with scaling, you may be using a driver that does not support scaling.

Window Menu Commands

The **New Window** command opens a new DB Monitor Expert frame window. So you can connect to a different Oracle database and monitor two or more databases simultaneously.

The **Cascade** command layers sheet windows inside DB Monitor Expert frame window so each title bar is visible. In addition, it arranges icons of minimized sheets across the bottom of the frame. To activate a sheet click sheet's title bar or use hot key SHIFT TAB to move between sheets.

The **Layer** command layers the sheets that are not minimized so that each sheet completely covers the one below it. In addition, it arranges icons of minimized sheets across the bottom of the frame. You can use hot key SHIFT TAB to move between sheets.

The **Tile Horizontally** command arranges sheet windows inside DB Monitor Expert frame window horizontally in equal sizes to fit on your screen. In addition, it arranges icons of minimized sheets across the bottom of the frame. To activate a sheet, click sheet's title bar.

The **Tile Vertically** command arranges sheet windows inside DB Monitor Expert frame window vertically in equal sizes to fit on your screen. In addition, it arranges icons of minimized sheets across the bottom of the frame. To activate a sheet, click sheet's title bar.

The **Arrange Icons** command arranges icons of minimized sheet windows across the bottom of the DB Monitor Expert frame window.

Table painter and table/view graphical design

To open Table painter select **Tools/Table Painter** command either in SQL Editor & Debugger or in DB Monitor Expert components (shortcut: CTRL T)

Working with objects in the Table Painter workspace

In the Table painter, you can:


- Graphically display tables and views in the database
- Graphically display [relationships diagrams](#)
- Open and close tables, including tables related to a selected table
- Drop tables and views
- Create, alter, and drop primary keys, foreign keys, and indexes
- Alter tables and views
- Add/modify comments for existing tables and views
- Extract tables and views definitions to the system clipboard
- Graphically select columns for [data manipulation](#)

Opening a table

When you first open the Table painter, the Select Tables dialog box displays listing all the tables and views in the current database that you have access to. You can create a new table or view, open existing tables and views or alter existing table or view.

To open a table or view:

1. Choose schema from **Owner** drop-down list.
2. Select one or more table/view names in the table list. To select multiple tables, hold down the CTRL key, and then click each object you want to select. To select range of objects, hold down the SHIFT key and then click on first and last object from the object list. To select all objects use shortcut CTRL +, to deselect all objects use shortcut CTRL -.
3. Click **OK** button or press ENTER key.

To add more tables and views to already open Table painter select **Objects/Table/Views...** command from the application menu or simply click  icon. The Select Tables dialog box will appear again. Repeat steps from 1 to 3.

Graphically selecting data from a table

Select one or more columns in the table/view columns list and then click **Objects/Data Manipulation...** command, or right-click the table. The table popup menu will appear. Select **Data Manipulation...** command. DB Monitor Expert will create data grid for selected columns and retrieves data from the database. See [Working with data](#) topic for details. Note that if the table has the primary key or unique constraint then the result set will be updateable. The result set from a view never will be updateable.

Moving objects

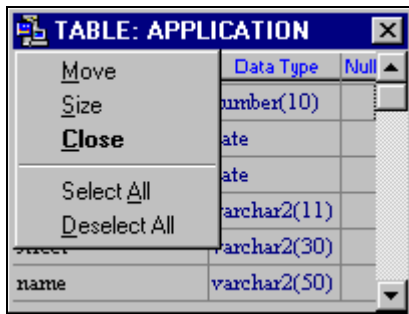
You can move an object around the workspace by dragging it with the mouse. For example, to move a table, press the left mouse button on the title bar for the table and drag the table, to move an index press the left mouse button on the index icon and drag the icon.

Resizing objects

You can resize an object in the workspace by dragging one of its corners or alternatively select **Size** command from the system menu of the object and use keyboard navigation keys to change object size.

System menu

To invoke objects system menu click on the icon in upper left corner of the table. You can also select following menu items from the table system menu



Move command - use this command to move the object by using keyboard

Size command - use this command to resize the object by using keyboard

Close command - use this command to close the object and its child icons such as index and referential integrity constraints.

Select All - use this command to select all columns in the table for data manipulation. See [data manipulation](#) topic for details.

Deselect All - use this command to deselect all previously selected columns.

Object popup menu

When you point at the table/view name or index/key icon and press the right mouse button, you display the popup menu for the object. The contents of this menu is different for different kind of objects

- Table popup menu

Command	Description
Table Design	Opens table design dialog
Drop Table	Drops table
Data Manipulation	Creates data grid for selected columns and retrieves data from the database. See data manipulation topic for details. Note that if the table has the primary key or unique constraint then the result set will be updateable.
Export Syntax To Log	Opens DDL export dialog and exports the specified table properties to the specified file. The export is appended to the end of the file
Zoom	Opens zoom dialog
Close	Closes selected table
Help	Invokes program help

- View popup menu

Command	Description
View Design	Opens view design dialog
Drop View	Drops view
Data Manipulation	Creates data grid for selected columns and retrieves data from the database. See data manipulation topic for details. Note that the result set will not be updateable.
Export Syntax To Clipboard	Opens DDL export dialog and exports specified view properties to the system clipboard
Zoom	Opens zoom dialog
Close	Closes selected view
Help	Invokes program help

- Referential key popup menu

Command	Description
Show Definition	Displays key definition in a message box.
Open Referenced Table (for foreign keys only)	Opens and expand the primary key table.
Open Dependent Table(s) (for primary keys only)	Opens and expands all tables in the database containing foreign keys that reference the selected primary key.

Drop Key	Drops key
Alter Key	Currently not supported
Help	Invokes program help

- Index popup menu

Command	Description
Show Definition	Displays index definition in a message box.
Drop Index	Drops index
Alter Index	Currently not supported
Help	Invokes program help

Modifying table properties

You can modify the properties of the table that is selected in the workspace. Use right mouse click on the table invoking table popup menu. From the menu select **Table Design** command. A [table design dialog](#) will appear.

Opening related tables

When working with tables containing keys, you can easily open related tables.

To open the table that a particular foreign key references:

1. Open and expand the table containing the foreign key.
2. Click the right mouse button on the icon representing the foreign key.
3. Select **Open Referenced Table** command from the popup menu. DB Monitor Expert will open and expand the primary key table.

To open all tables referencing a particular primary key:

1. Open and expand the table containing the primary key.
2. Click the right mouse button on the button representing the primary key.
3. Select **Open Dependent Table(s)** command from the popup menu. DB Monitor Expert will open and expands all tables in the database containing foreign keys that reference the selected primary key.

Closing a table

You can remove a table from the workspace. This action only removes the table graphical presentation from the Table painter workspace. It does not drop the table from the database.

To close a table:

1. Use right-click on the table. DB Monitor Expert will display the popup menu.
2. Select **Close** command from the popup menu.

or

1. Double click on the icon in the table/view left upper corner.

or

2. Click on the icon the table/view left upper corner. DB Monitor Expert will display objects system menu.
3. Select **Close** command from the system menu.

Keyboard alternatives

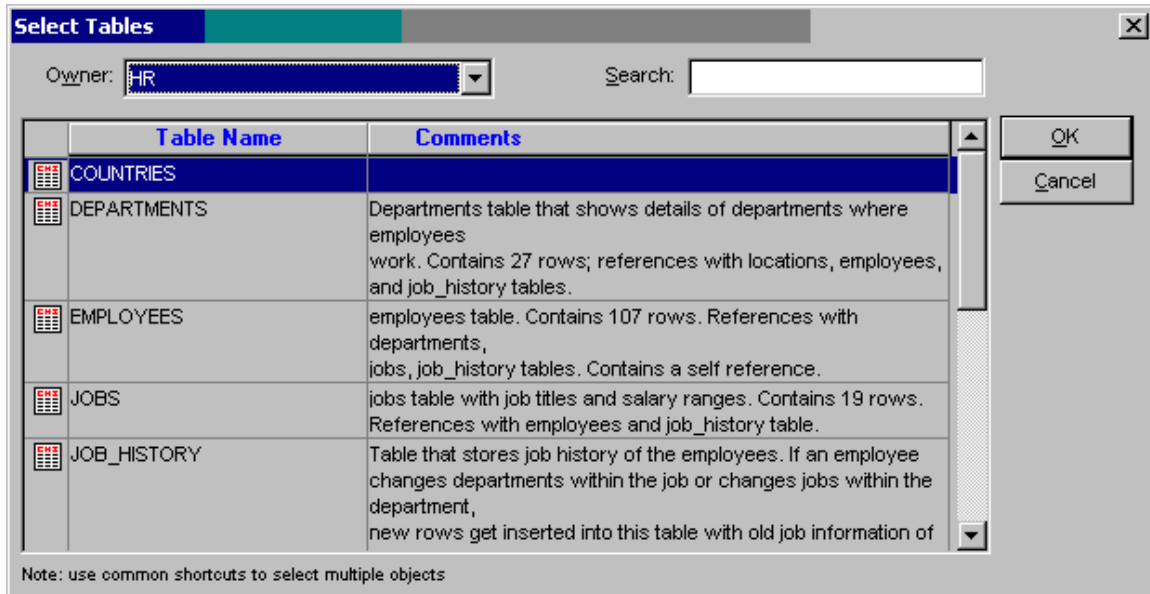
DB Monitor Expert provides keyboard alternatives to common workspace activities:

To	Do this
Select a table or view or key	Press TAB to move left to right among the opened tables and views, or press SHIFT+TAB to move right to left
Add new table or view to the Table painter	Press ALT+O, T
Browse an index or key	Press TAB or SHIFT+TAB to select the index or key to browse and press ENTER
Scroll in the painter workspace	If you open more tables and views than can be displayed in the Table painter workspace at one time, you can scroll up or down to view all the tables and views. To scroll up, press UP ARROW. To scroll down, press DOWN ARROW. In addition you can use PAGE UP and PAGE DOWN keys for fast scrolling.

See also:

- [Table relationships](#)
- [Quick data view and data manipulation](#)
- [DDL Extraction](#)
- [Table difference analyzer](#)
- [Table space estimation](#)
- [Data analysis - data histogram](#)
- [Index analysis - index histogram](#)

Select Table(s) dialog



This dialog used to select one or multiple tables, and whenever applicable views and synonyms and return the selected to the previous screen. Current selection is always highlighted.

To quickly locate the required object choose object's owner name from the **Owner** drop-down list. This will set a filter by the schema name and reduce the number of listed objects. You can also use the **Search** field to locate the object. As you type object name in the field DB Tools will automatically scroll the list and highlight the nearest match.

Tips:

- To select a single object single click the required object. Click the **OK** button to close the dialog and return to the previous screen.
- To select a range of objects listed one after another click the first object, then while holding down the SHIFT key click the last object. All objects in the selected range will become highlighted. If necessary you can use the mouse to scroll the table list while or before holding down the SHIFT key. Do not use the **Owner** and **Search** options at this time. Click the **OK** button to close the dialog and return to the previous screen.
- To select non-consecutive objects hold down the click on the first object, then while holding down the CTRL key click all required objects. All clicked objects will become highlighted. If necessary you can use the mouse to scroll the table list while or before holding down the CTRL key. Do not use the **Owner** and **Search** options at this time. Click the **OK** button to close the dialog and return to the previous screen.
- To quickly select/unselect all objects in the list use CTRL + and CTRL – keyboard shortcuts. If you have a filter set using the **Owner** drop-down the selected will apply to visible objects names only.

Table Design

Overview

Use the Table Design dialog to create and alter database tables by defining columns, column comments, and table constraints. In addition, you can use that dialog to rename database tables. You can open Table

Design dialog window from [Table painter](#).

Table Design dialog elements

- **Name:** Type the name of the table you want to create or a new name for the opened table.
- **Owner:** Select the owner for the table from the user names in the drop-down list. Note that you cannot change the owner for already existing table.
- **Table definition grid:** Each row of the table definition grid represents a column in the table you are creating or altering. The grid contains the following columns:

Column Name

Enter the name of the column. Each column name must be unique within the table. The maximum name length is limited to 30 characters.

Type

Click the Type column to select a valid Oracle data type from the drop down list. Type is the type of data that will be placed in this column. For example, the data type NUMBER is used for a column in which the data will always be numerical. The data type for an address column would be CHAR because this will contain both numerical and alphabetical characters.

Size

Enter the number of characters or bytes necessary to hold your data. For NUMBER data types, data size includes the total number of digits on both the left and right side of the decimal point.

Scale

For NUMBER data types, enter the number of digits to the right of the decimal point.

Default

This is the default value for the column that should be entered in case none is specified. This feature is useful if a column cannot be [null](#).

Not Null

Check this box to specify if this column cannot be [null](#).

Unique

Check this option if the column stores unique values only. DB Monitor Expert will create unique constraint for this column.

Primary Key

A set of columns used to enforce uniqueness of rows. The combination of column values is unique for each row in the table. The primary key is the most frequently used means of accessing rows. This field is also used to sort your table and to keep you from duplicating records. Primary keys can be made up of a single column or several columns and are used to locate specific rows of data. An example of a primary key would be a column in a table for a product ID number as long as no two products have the same ID number.

To graphically design the primary key, click on the **Primary Key** button to start [Primary Key Definition](#) dialog. Note that order in which you check primary key columns is significant.

The number in the first field designates the position of this column in the primary key that uniquely identifies a row in the table.

Foreign Key

Click **Foreign Key** button to start [Foreign Key Browser](#) from which you can select existing keys, drop them or create new ones. Use the **New** button on [Foreign Key Browser](#) to open [Foreign Key Definition dialog](#)

Check Condition

Enter a combination of one or more expressions and logical operators that evaluate to TRUE, FALSE, or unknown. You can make entries here when creating a new table. However, once a table has been created, you cannot edit the check condition of existing columns. An example of a check condition statement follows:

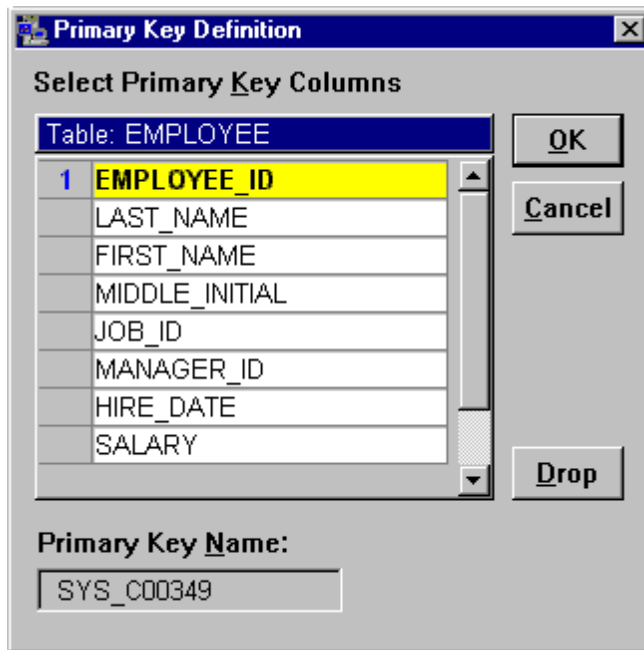
```
ORDER_ID IS NOT NULL
ORDER_DATE < SYSDATE
TOTAL>=0
```

Comments

Enter any comments for this column.

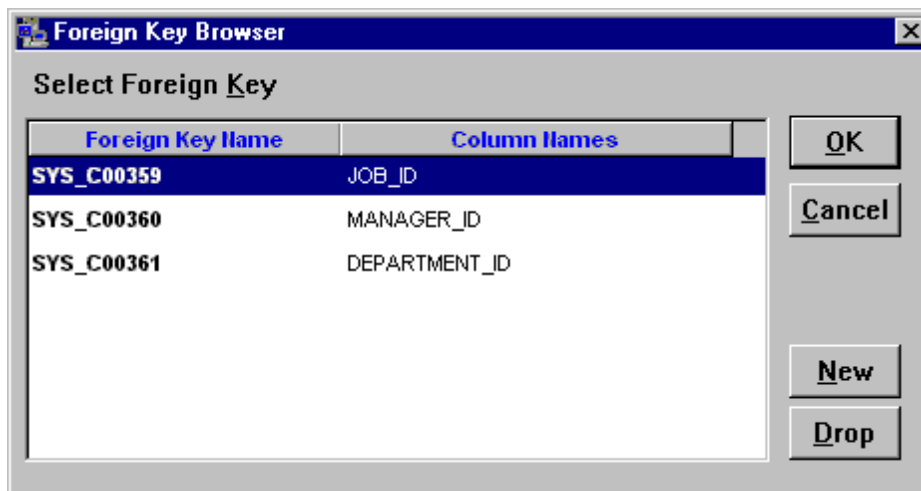
- **Insert** button: Click this button to insert new row before current in the definition grid. Note that this button is available only for the new table that has not been created yet.
- **Add** button: Click this button to add new row at the end of the definition grid. Note that this button is available only for the table that has been created already.
- **Delete** button: Click this button to remove the current row from the definition grid. However you may not remove columns that have been created already.
- **Primary Key** button: To graphically design the primary key, click on the **Primary Key** button to start [Primary Key Definition](#) dialog. Note that order in which you check primary key columns is significant.
- **Foreign Key** button: Click this button to start [Foreign Key Browser](#) from which you can select existing keys, drop them or create new ones. Use the **New** button on [Foreign Key Browser](#) to open **Foreign Key Definition** dialog.
- **OK** button: Click this button to save changes in the database. This will attempt to create a new table or alter existing table, which is being opened in the Table Design dialog.
- **Cancel** button: Click this button to close the dialog and cancel all changes if any.

Primary Key Design

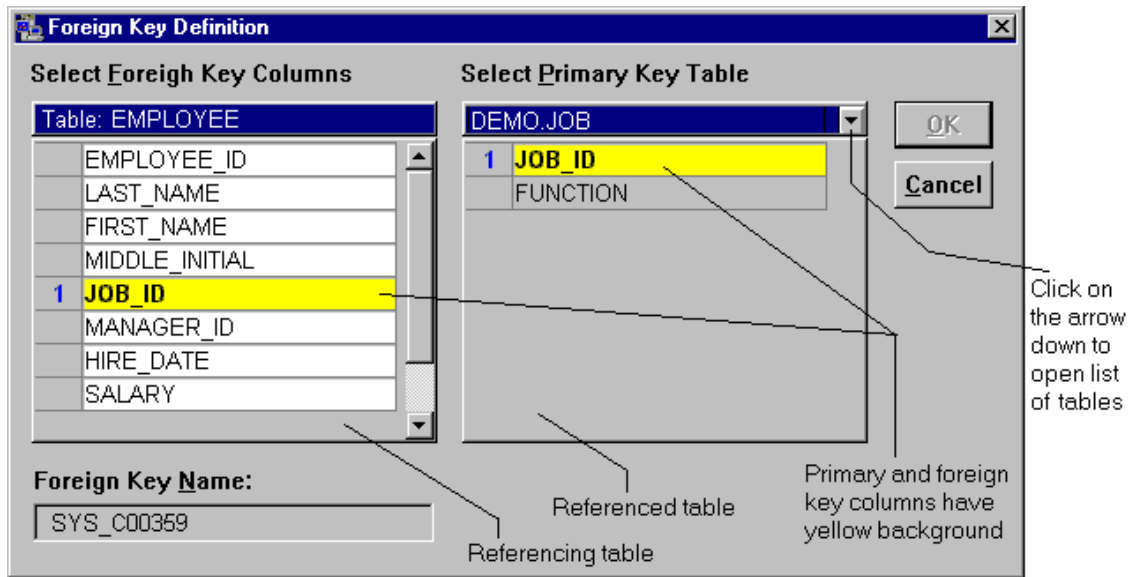


The number in the first field designates the position of this column in the primary key that uniquely identifies a row in the table.

Foreign Key Browser



Foreign Key Dialog



The number in the first field of the primary key table designates the position of this column in the primary key that uniquely identifies a row in the table.

The number in the first field of the foreign key table designates the position of this column in the foreign key that must match position of matching column in the referenced primary key.

DDL Reverse-Engineering

Both SQL Editor and DB Monitor Expert can be used to reverse-engineer tables, views and most other database objects. They support a number of alternative reverse-engineer methods. Do any of the following to extract table or view definition:

1. Open table or view you want to reverse-engineer in the [Table painter](#).
2. Select **Objects/Export Syntax to Clipboard** command

or

1. Open Database Navigator tool using Tools/Database Navigator menu or simply by pressing F7 Shortcut.
2. Navigate to schema and table you want to reverse-engineer. Click on the Code tab page. To customize which options are included in the generated script use global system Options dialog (Tools/Options menu); [DDL Export Options](#) Page. See [Customization](#) topic for detail.

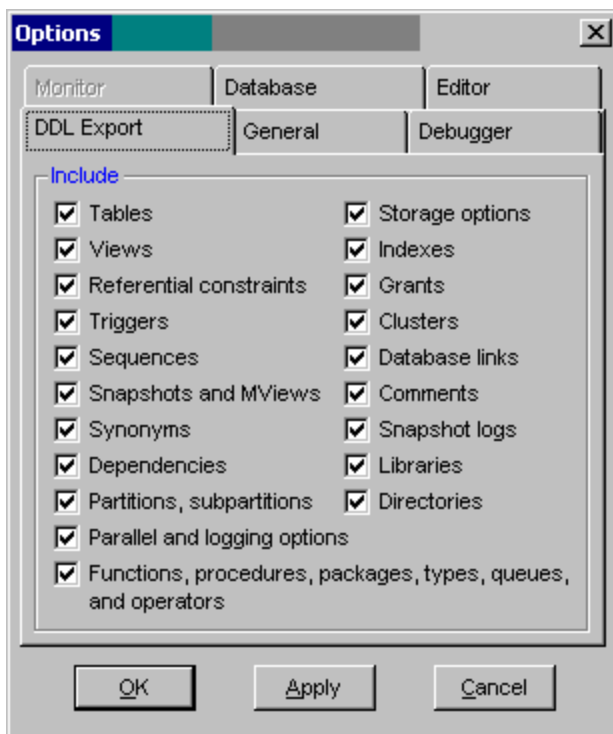
or

1. Open Database Navigator tool using Tools/Database Navigator menu or simply by pressing F7 Shortcut.
2. Navigate to schema and table you want. Right-click the object and then select **Generate and Copy/CREATE** command in the [popup menu](#). Click on the Code tab page. To customize which options are included in the generated script use global system Options dialog (Tools/Options menu).

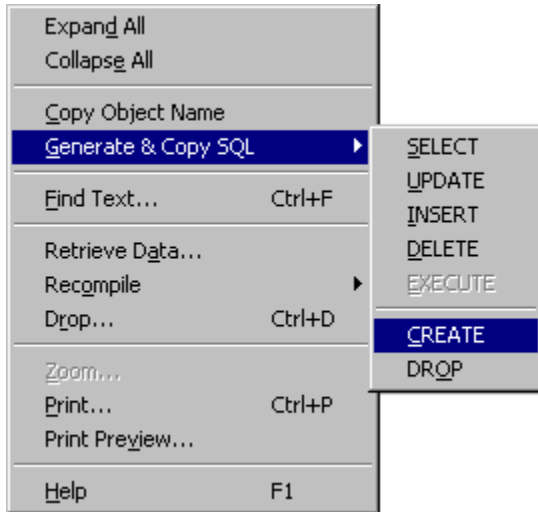
or

1. Open that table or view in the [Table painter](#). Select **Objects/Export Syntax to Clipboard** command (Shortcut: ALT O, S) from the program menu. Alternatively, you can right-click on the table or view graphical presentation. DB Monitor Expert will display the popup menu from which you can select **Export Syntax To Clipboard** command. The [DDL Export Options](#) dialog will appear.
2. Choose which options are included in the generated script. Note that the [DDL Export Options](#) dialog options default to options specified in the program preferences. See [Customization](#) topic for detail.
3. Click **OK** button. The extracted DDL will be exported to the system clipboard and the clipboard viewer will be launched. If you want you can paste the clipboard contents to the [DBA Notepad](#) or another application.

DDL Reverse-Engineering Options



Reverse-engineering DDL Using Popup Menu



1. In Database Object Navigator select the object you want to reverse-engineer.
2. Right-click the object icon or name displayed in the left-side tree-view.
3. From the Popup Menu select **Generate & copy SQL/CREATE** command. This will copy reverse-engineered SQL commands required to create the selected object.

View Design Dialog

View design dialog is used for creation and/or alteration of the database views.

To create new database view

1. To start View design dialog launch [Table painter](#) first. Select view owner from the **Owner** drop-down list. DB Monitor Expert will display list of available database tables and view. Click on the **New View** button. New view dialog box will appear.
2. Type name of the view in the **Name** field.
3. Type or paste from the clipboard, SQL select statement for the view. Make sure that the SQL statement does not end with a semicolon (;).
4. Click **Create** button.

To alter existing database view

1. To start View design dialog launch [Table painter](#) first. Select view owner from the **Owner** drop-down list. DB Monitor Expert will display list of available database tables and view. Highlight required view in the **Table/View** list. Click on the **Alter View** button. Alter view dialog box will appear.
2. Type or paste from the clipboard, SQL select statement for the view. Make sure that the SQL statement does not end with a semicolon (;).
3. Click **Alter** button.

Functions in Sort and Filter Expressions

Functions by category

You can use the following functions in sort and filter expressions. The functions are organized in the following categories.

[Aggregate](#)
[Crosstab](#)
[Data type checking and conversion functions](#)
[Day, date, and time functions](#)
[Miscellaneous functions](#)
[Numeric functions](#)
String functions

See also:

[Operators](#)

Aggregate functions

Do not use the following functions in sort and filter expressions. They are not supported in such expressions.

Avg
Count
CumulativePercent
CumulativeSum
First
Large
Last
Max
Median
Min
Mode
Percent
Small
StDev
StDevP
Sum
Var
VarP

See also:

[Functions by category](#)

Crosstab functions

You can use the following functions only in user-defined reports that have **Crosstab** presentation style.

[CrosstabAvg](#)
[CrosstabCount](#)
[CrosstabMax](#)
[CrosstabMin](#)
[CrosstabSum](#)

See also:[Functions by category](#)**Data type checking and conversion functions**[Asc](#)[Char](#)[Date](#)[DateTime](#)[Integer](#)[IsDate](#)[IsNull](#)[IsNumber](#)[IsTime](#)[Long](#)[Number](#)[String](#)[Time](#)**See also:**[Functions by category](#)**Date and time functions**[Date](#)[DateTime](#)[Day](#)[DayName](#)[DayNumber](#)[DaysAfter](#)[Hour](#)[Minute](#)[Month](#)[Now](#)[RelativeDate](#)[RelativeTime](#)[Second](#)[SecondsAfter](#)[Time](#)[Today](#)[Year](#)**See also:**[Functions by category](#)**Miscellaneous functions**[Case](#)[If](#)[ProfileInt](#)[ProfileString](#)[RGB](#)

See also:[Functions by category](#)**Numeric functions**[Abs](#)[Ceiling](#)[Cos](#)[Exp](#)[Fact](#)[Int](#)[Log](#)[LogTen](#)[Mod](#)[Pi](#)[Rand](#)[Round](#)[Sign](#)[Sin](#)[Sqrt](#)[Tan](#)[Truncate](#)**See also:**[Functions by category](#)**String functions**[Fill](#)[Left](#)[LeftTrim](#)[Len](#)[Lower](#)[Match](#)[Mid](#)[Pos](#)[Replace](#)[Right](#)[RightTrim](#)[Space](#)[String](#)[Trim](#)[Upper](#)[WordCap](#)**See also:**[Functions by category](#)**Asc****Description:** Converts the first character of a string to its ASCII integer value.**Syntax:** [Asc](#) ([string](#))

Argument	Description
string	The string for which you want the ASCII value of the first character

Return value: Integer. Returns the ASCII value of the first character in [string](#).

Usage: Use Asc to test the case of a character or manipulate text and letters. To find out the case of a character, you can check whether its ASCII value is within the appropriate range.

Char

Description: Converts an integer to a character.

Syntax: [Char \(n \)](#)

Argument	Description
n	The integer you want to convert to a character

Return value: String. Returns the character whose ASCII value is [n](#).

Integer

Description: Converts the value of a string to an integer.

Syntax: [Integer \(string \)](#)

Argument	Description
string	The string you want returned as an integer

Return value: Integer. Returns the contents of [string](#) as an integer if it succeeds and 0 if [string](#) is not a number.

IsDate

Description: Tests whether a string value is a valid date.

Syntax: [IsDate \(string \)](#)

Argument	Description
string	A string whose value you want to test to determine whether it is a valid date

Return value: Boolean. Returns TRUE if [datevalue](#) is a valid date and FALSE if it is not.

IsNull

Description: Reports whether the value of a column or expression is NULL.

Syntax: `IsNull (any)`

Argument	Description
<code>any</code>	A column or expression that you want to test to determine whether its value is NULL

Return value: Boolean. Returns TRUE if `any` is NULL and FALSE if it is not.

Usage: Use `IsNull` to test whether a user-entered value or a value retrieved from the database is NULL.

IsNumber

Description: Reports whether the value of a string is a number.

Syntax: `IsNumber (string)`

Argument	Description
<code>string</code>	A string whose value you want to test to determine whether it is a valid number

Return value: Boolean. Returns TRUE if `string` is a valid number and FALSE if it is not.

IsTime

Description: Reports whether the value of a string is a valid time value.

Syntax: `IsTime (timevalue)`

Argument	Description
<code>timevalue</code>	A string whose value you want to test to determine whether it is a valid time

Return value: Boolean. Returns TRUE if `timevalue` is a valid time and FALSE if it is not.

Long

Description: Converts the value of a string to a long.

Syntax: `Long (string)`

Argument	Description
<code>string</code>	The string you want returned as a long

Return value: Long. Returns the contents of `string` as a long if it succeeds and 0 if `string` is not a valid number.

Number

Description: Converts a string to a number.

Syntax: `Number (string)`

Argument	Description
<code>string</code>	The string you want returned as a number

Return value: A numeric data type. Returns the contents of `string` as a number. If `string` is not a valid number, Number returns 0.

GetRow

Description: Reports the number of a row in which processing is performed.

Syntax: `GetRow ()`

Return value: Long. Returns the number of a row if it succeeds, 0 if no data has been retrieved or added, and -1 if an error occurs.

IsRowModified

Description: Reports whether the row has been modified.

Syntax: `IsRowModified ()`

Return value: Boolean. Returns TRUE if the row has been modified and FALSE if it has not.

Example: To filter out only modified rows, set filter expression as following:
`IsRowModified ()`

IsRowNew

Description: Reports whether the row has been newly inserted after last save or retrieve.

Syntax: `IsRowNew ()`

Return value: Returns TRUE if the row is new and FALSE if it was retrieved from the database.

Example: To filter out only newly inserted rows, set filter expression as following:
`IsRowNew ()`

Date

Description: Converts a string whose value is a valid date to a value of data type date.

Syntax: `Date (string)`

Argument	Description
<code>string</code>	A string containing a valid date (such as Jan 1, 1998, or 12-31-99) that you want returned as a date

Return value: Date. Returns the date in **string** as a date. If **string** does not contain a valid date, Date returns NULL.

Usage: The value of the string must be a valid date.

Valid dates can include any combination of day (1-31), month (1-12 or the name or abbreviation of a month), and year (two or four digits). Leading zeros are optional for month and day. If the month is a name or an abbreviation, it can come before or after the day; if it is a number, it must be in the month location specified in the Windows control panel. A four-digit number is assumed to be a year.

If the year is two digits, then program chooses the century, as follows. If the year is between 00 and 49, program assumes 20 as the first two digits; if it is between 50 and 99, DB Monitor Expert assumes 19. If your data includes dates before 1950, such as birth dates, always specify a four-digit year so that DB Monitor Expert interprets the date as intended.

DB Tools handles years from 1000 to 3000 inclusive.

An expression has a more limited set of data types than the functions that can be part of the expression. Although the Date function returns a date value, the whole expression is promoted to a DateTime value. Therefore, if your expression consists of a single Date function, it will appear that Date returns the wrong data type. To convert the date without the time, choose an appropriate convert format.

Time

Description: Converts a string to a time data type.

Syntax: **Time** (**string**)

Argument	Description
string	A string containing a valid time (such as 8AM or 10:25) that you want returned as a time data type. Only the hour is required; you do not have to include the minutes, seconds, or microseconds of the time or AM or PM. The default value for minutes and seconds is 00 and for microseconds is 000000. AM or PM is determined automatically

Return value: Time. Returns the time in **string** as a time data type. If **string** does not contain a valid time, Time returns 00:00:00.

DateTime

Description: Combines a date and a time value into a DateTime value.

Syntax: **DateTime** (**date** {, **time**})

Argument	Description
date	A valid date (such as Jan 1, 1998, or 12-31-99) or a blob variable whose first value is a date that you want included in the value returned by DateTime
time (optional)	(optional) A valid time (such as 8AM or 10:25:23:456799) or a blob variable whose first value is a time that you want included in the value returned by DateTime. If you include a time, only the hour portion is required. If you omit the minutes, seconds, or microseconds, they are assumed to be 0s. If you omit AM or PM, the hour is determined

according to the 24-hour clock

Return value: DateTime. Returns a DateTime value based on the values in [date](#) and optionally [time](#). If [time](#) is omitted, DateTime uses 00:00:00.000000 (midnight).

Usage: For information on valid dates, see [Date](#) function.

Day

Description: Obtains the day of the month in a date value.

Syntax: [Day](#) ([date](#))

Argument	Description
date	The date from which you want the day

Return value: Integer. Returns an integer (1-31) representing the day of the month in [date](#).

DayName

Description: Determines the day of the week in a date value and returns the weekday's name.

Syntax: [DayName](#) ([date](#))

Argument	Description
date	The date for which you want the name of the day

Return value: String. Returns a string whose value is the name of the weekday (Sunday, Monday, and so on) for [date](#).

DayNumber

Description: Determines the day of the week of a date value and returns the number of the weekday.

Syntax: [DayNumber](#) ([date](#))

Argument	Description
date	The date from which you want the number of the day of the week

Return value: Integer. Returns an integer (1-7) representing the day of the week of [date](#). Sunday is day 1, Monday is day 2, and so on.

DaysAfter

Description: Determines the day of the week of a date value and returns the number of the weekday.

Syntax: [DaysAfter](#) ([date1](#), [date2](#))

Argument	Description
date1	A date value that is the start date of the interval being measured
date2e	A date value that is the end date of the interval

Return value: Long. Returns a long containing the number of days [date2](#) occurs after [date1](#). If [date2](#) occurs before [date1](#), DaysAfter returns a negative number.

Hour

Description: Obtains the hour in a time value. The hour is based on a 24-hour clock.

Syntax: [Hour \(time \)](#)

Argument	Description
time	The time value from which you want the hour

Return value: Integer. Returns an integer (00-23) containing the hour portion of [time](#).

Minute

Description: Obtains the number of minutes in the minutes portion of a time value.

Syntax: [Minute \(time \)](#)

Argument	Description
time	The time value from which you want the minutes

Return value: Integer. Returns the minutes portion of [time](#) (00 to 59).

Month

Description: Determines the month of a date value.

Syntax: [Month \(date \)](#)

Argument	Description
date	The date from which you want the month

Return value: Integer. Returns an integer (1 to 12) whose value is the month portion of [date](#).

Now

Description: Obtains the current time based on the system time of the client machine.

Syntax: `Now ()`

Return value: Time. Returns the current time based on the system time of the client machine.

RelativeDate

Description: Obtains the date that occurs a specified number of days after or before another date.

Syntax: `RelativeDate (date, n)`

Argument	Description
<code>date</code>	A date value
<code>n</code>	An integer indicating the number of days

Return value: Date. Returns the date that occurs `n` days after `date` if `n` is greater than 0. Returns the date that occurs `n` days before `date` if `n` is less than 0.

RelativeTime

Description: Obtains the time that occurs a specified number of seconds after or before another time.

Syntax: `RelativeTime (time, n)`

Argument	Description
<code>time</code>	A time value
<code>n</code>	A long number of seconds

Return value: Time. Returns the time that occurs `n` seconds after `time` if `n` is greater than 0. Returns the time that occurs `n` seconds before `time` if `n` is less than 0. The maximum return value is 23:59:59.

Second

Description: Obtains the number of seconds in the seconds portion of a time value.

Syntax: `Second (time)`

Argument	Description
<code>time</code>	The time value from which you want the seconds

Return value: Integer. Returns the seconds portion of time (00 to 59).

SecondsAfter

Description: Determines the number of seconds one time occurs after another.

Syntax: `SecondsAfter (time1, time2)`

Argument	Description
time1	A time value that is the start time of the interval being measured
time2	A time value that is the end time of the interval

Return value: Long. Returns the number of seconds [time2](#) occurs after [time1](#). If [time2](#) occurs before [time1](#), SecondsAfter returns a negative number.

Today

Description: Obtains the system date and time.

Syntax: [Today \(\)](#)

Return value: DateTime. Returns the current system date and time.

Year

Description: Determines the year of a date value.

Syntax: [Year \(date \)](#)

Argument	Description
date	The date from which you want the year

Return value: Integer. Returns an integer whose value is a four-digit year adapted from the year portion of [date](#) if it succeeds and 1900 if an error occurs.

If the year is two digits, then program chooses the century, as follows. If the year is between 00 to 49, program assumes 20 as the first two digits; if it is between 50 and 99, program assumes 19.

Usage: Obtains the year portion of date. DB Monitor Expert handles years from 1000 to 3000 inclusive. If your data includes dates before 1950, such as birth dates, always specify a 4-digit year so that Year function (and other functions) interprets the date as intended.

Case

Description: Tests the values of a column or expression and returns values based on the results of the test.

Syntax:

[Case \(column WHEN value1 THEN result1 { WHEN value2 THEN result2 { ...} } { ELSE resultelse} \)](#)

Argument	Description
column	The column or expression whose values you want to test. Column can be the column name or the column number preceded by a pound sign (#). Column can also be an expression that includes a reference to the column. Column is compared to each valuen
WHEN	Introduces a value-result pair. At least one WHEN is required
valuen	One or more values that you want to compare to values of column. A value can be:

- A single value
- A list of values separated by commas (for example, 2, 4, 6, 8)
- A TO clause (for example, 1 TO 20)
- IS followed by a relational operator and comparison value (for example, IS>5)
- Any combination of the above with an implied OR between expressions (for example, 1,3,5,7,9,27 TO 33, IS>42)

THEN Introduces the result to be returned when column matches the corresponding **valuen**

resultn An expression whose value is returned by Case for the corresponding **valuen**. All **resultn** must have the same data type

ELSE (optional) (optional) Specifies that for any values of column that don't match the values of **valuen** already specified, Case returns **resultelse**

resultelse An expression whose value is returned by Case when the value of column doesn't match any WHEN **valuen** expression

Return value: The data type of **resultn**. Returns the result you specify in **resultn**.

Usage: If more than one WHEN clause matches column, Case returns the result of the first matching one.



Note:

Function Case is similar to the ORACLE's Decode function.

If

Description: Evaluates a condition and returns a value based on that condition.

Syntax: If (**boolean**, **truevalue**, **falsevalue**)

Argument	Description
boolean	A boolean expression that evaluates to TRUE or FALSE
truevalue	A string containing the value you want returned if the boolean expression is TRUE
falsevalue	A string containing the value you want returned if the boolean expression is FALSE

Return value: The data type of **truevalue** or **falsevalue**. Returns **truevalue** if **boolean** is TRUE and **falsevalue** if it is FALSE. Returns NULL if an error occurs.

ProfileInt

Description: Obtains the integer value of a setting in the specified profile file.

Syntax: ProfileInt (**filename**, **section**, **key**, **default**)

Argument	Description
----------	-------------

filename	A string whose value is the name of the profile file. If you do not specify a full path, ProfileInt uses the operating system's standard file search order to find the file
section	A string whose value is the name of a group of related values in the profile file. In the file, section names are in square brackets. Do not include the brackets in section. Section is not case-sensitive
key	A string whose value is the name of a group of related values in the profile file. In the file, section names are in square brackets. Do not include the brackets in section . Section is not case-sensitive
default	An integer value that ProfileInt will return if filename is not found, if section or key does not exist in filename , or if the value of key cannot be converted to an integer

Return value: Integer. Returns default if [filename](#) is not found, [section](#) is not found in [filename](#), or [key](#) is not found in [section](#), or the value of [key](#) is not an integer. Returns -1 if an error occurs.

Usage: Use ProfileInt or [ProfileString](#) to get configuration settings from a profile file.

ProfileString

Description: Obtains the string value of a setting in the specified profile file.

Syntax: ProfileInt ([filename](#), [section](#), [key](#), [default](#))

Argument	Description
Filename	A string whose value is the name of the profile file. If you do not specify a full path, ProfileString uses the operating system's standard file search order to find the file
Section	A string whose value is the name of a group of related values in the profile file. In the file, section names are in square brackets. Do not include the brackets in section. Section is not case-sensitive
Key	A string whose value is the name of a group of related values in the profile file. In the file, section names are in square brackets. Do not include the brackets in section . Section is not case-sensitive
default	An string value that ProfileString will return if filename is not found, if section or key does not exist in filename

Return value: String, with a maximum length of 4096 characters. Returns the string from [key](#) within [section](#) within [filename](#). If [filename](#) is not found, [section](#) is not found in [filename](#), or [key](#) is not found in [section](#), ProfileString returns [default](#). If an error occurs, it returns the empty string ("").

Usage: Use [ProfileInt](#) or ProfileString to get configuration settings from a profile file.

RGB

Description: Calculates the long value that represents the color specified by numeric values for the red, green, and blue components of the color.

Syntax: `RGB (red, green, blue)`

Argument	Description
<code>red</code>	The integer value of the red component of the required color
<code>green</code>	The integer value of the green component of the required color
<code>blue</code>	The integer value of the blue component of the required color

Return value: Long. Returns the long that represents the color created by combining the values specified in `red`, `green`, and `blue`. If an error occurs, RGB returns NULL.

Usage: The formula for combining the colors is:

$\text{Red} + (256 * \text{Green}) + (65536 * \text{Blue})$

Use RGB to obtain the long value required to set the color for text and drawing objects. You can also set an object's color to the long value that represents the color. The RGB function provides an easy way to calculate that value. The value of a component color is an integer between 0 and 255

Abs

Description: Calculates the absolute value of a number.

Syntax: `Abs (n)`

Argument	Description
<code>n</code>	The number for which you want the absolute value

Return value: The data type of `n`. Returns the absolute value of `n`.

Ceiling

Description: Determines the smallest whole number that is greater than or equal to a specified limit.

Syntax: `Ceiling (n)`

Argument	Description
<code>N</code>	The number for which you want the smallest whole number that is greater than or equal to it

Return value: The data type of `n`. Returns the smallest whole number that is greater than or equal to `n`.

Cos

Description: Calculates the cosine of an angle.

Syntax: `Cos (n)`

Argument	Description
<code>n</code>	The angle (in radians) for which you want the cosine

Return value: Double. Returns the cosine of `n`.

Sin

Description: Calculates the sine of an angle.

Syntax: `Sin (n)`

Argument	Description
<code>n</code>	The angle (in radians) for which you want the sine

Return value: Double. Returns the sine of `n`.

Exp

Description: Raises `e` to the specified power.

Syntax: `Exp (n)`

Argument	Description
<code>n</code>	The power to which you want to raise <code>e</code> (2.71828)

Return value: Double. Returns `e` raised to the power `n`.

Fact

Description: Determines the factorial of a number.

Syntax: `Fact (n)`

Argument	Description
<code>n</code>	The number for which you want the factorial

Return value: Double. Returns the factorial of `n`.

Int

Description: Determines the largest whole number less than or equal to a number.

Syntax: `Int (n)`

Argument	Description
n	The number for which you want the largest whole number that is less than or equal to it

Return value: The data type of [n](#). Returns the largest whole number less than or equal to [n](#).

Log

Description: Determines the natural logarithm of a number.

Syntax: [Log \(n \)](#)

Argument	Description
n	The number for which you want the natural logarithm (base e). The value of n must be greater than 0

Return value: Double. Returns the natural logarithm of [n](#). An execution error occurs if [n](#) is negative or zero.

Inverse: The inverse of the Log function is the [Exp](#) function.

LogTen

Description: Determines the base 10 logarithm of a number.

Syntax: [LogTen \(n \)](#)

Argument	Description
n	The number for which you want the base 10 logarithm. The value of n must not be negative

Return value: Double. Returns the decimal log of [n](#).

Obtaining a number: The expression $10^{\text{LogTen}(n)}$ is the inverse for [LogTen\(n\)](#). To obtain [n](#) given number ($\text{nbr} = \text{LogTen}(n)$), use $n = 10^{\text{nbr}}$.

Mod

Description: Obtains the remainder (modulus) of a division operation.

Syntax: [Mod \(x, y \)](#)

Argument	Description
x	The number you want to divide by y
y	The number you want to divide into x

Return value: The data type of [x](#) or [y](#), whichever data type is more precise.

Pi

Description: Multiplies pi by a specified number.

Syntax: `Pi (n)`

Argument	Description
<code>n</code>	The number you want to multiply by pi (3.14159265358979323...)

Return value: Double. Returns the result of multiplying `n` by pi if it succeeds and -1 if an error occurs.

Usage: Use `Pi` to convert angles to and from radians.

Rand

Description: Obtains a random whole number between 1 and a specified upper limit.

Syntax: `Rand (n)`

Argument	Description
<code>n</code>	The upper limit of the range of random numbers you want returned. The lower limit is always 1. The upper limit cannot exceed 32,767

Return value: A numeric data type, the data type of `n`. Returns a random whole number between 1 and `n`.

Usage: The sequence of numbers generated by repeated calls to the `Rand` function is a computer-generated pseudo-random sequence.

Round

Description: Rounds a number to the specified number of decimal places.

Syntax: `Round (x, n)`

Argument	Description
<code>x</code>	The number you want to round
<code>n</code>	The number of decimal places to which you want to round <code>x</code>

Return value: Decimal. If `n` is positive, returns `x` rounded to the specified number of decimal places. If `n` is negative, returns `x` rounded to $(-n+1)$ places before the decimal point. Returns -1 if it fails.

Sign

Description: Determines the sign of a number. `Sign` reports whether a number is negative, zero, or positive.

Syntax: `Sign (n)`

Argument	Description
n	The number for which you want to determine the sign

Return value: Integer. Returns a number (-1, 0, or 1) indicating the sign of [n](#).

Sqrt

Description: Calculates the square root of a number.

Syntax: [Sqrt \(n \)](#)

Argument	Description
n	The number for which you want the square root

Return value: Double. Returns the square root of [n](#).

Usage: [Sqrt\(n\)](#) is the same as $n^{.5}$.

Taking the square root of a negative number causes an execution error.

Tan

Description: Calculates the tangent of an angle.

Syntax: [Tan \(n \)](#)

Argument	Description
n	The angle (in radians) for which you want the tangent

Return value: Double. Returns the tangent of [n](#) if it succeeds and -1 if an error occurs.

Truncate

Description: Truncates a number to the specified number of decimal places.

Syntax: [Truncate \(x, n \)](#)

Argument	Description
x	The number you want to truncate
n	The number of decimal places to which you want to truncate x

Return value: The data type of [n](#). If [n](#) is positive, returns [x](#) truncated to the specified number of decimal places. If [n](#) is negative, returns [x](#) truncated to $(-n+1)$ places before the decimal point. Returns -1 if it fails.

Fill

Description: Builds a string of the specified length by repeating the specified characters until the result string is long enough.

Syntax: `Fill (chars, n)`

Argument	Description
<code>chars</code>	A string whose value will be repeated to fill the return string
<code>n</code>	A long whose value is the length of the string you want returned (from 0 to 60000)

Return value: String. Returns a string `n` characters long filled with repetitions of the characters in the argument `chars`. If the argument `chars` has more than `n` characters, the first `n` characters of `chars` are used to fill the return string. If the argument `chars` has fewer than `n` characters, the characters in `chars` are repeated until the return string has `n` characters.

Left

Description: Obtains a specified number of characters from the beginning of a string.

Syntax: `Left (string, n)`

Argument	Description
<code>string</code>	The string containing the characters you want
<code>n</code>	A long specifying the number of characters you want

Return value: String. Returns the leftmost `n` characters in `string` if it succeeds and the empty string ("") if an error occurs. If `n` is greater than or equal to the length of the `string`, `Left` returns the entire `string`. It does not add spaces to make the return value's length equal to `n`.

RightTrim

Description: Removes spaces from the end of a string.

Syntax: `RightTrim (string)`

Argument	Description
<code>string</code>	The string you want returned with trailing blanks deleted

Return value: String. Returns a copy of `string` with trailing blanks deleted if it succeeds and the empty string ("") if an error occurs.

LeftTrim

Description: Removes leading spaces from the beginning of a string.

Syntax: `LeftTrim (string)`

Argument	Description
string	The string you want returned with leading blanks deleted

Return value: String. Returns a copy of [string](#) with leading blanks deleted if it succeeds and the empty string ("") if an error occurs.

Len

Description: Reports the length of a string.

Syntax: [Len \(string \)](#)

Argument	Description
string	The string for which you want the length

Return value: Long. Returns a long containing the length of string if it succeeds and -1 if an error occurs.

Lower

Description: Converts all the characters in a string to lowercase.

Syntax: [Lower \(string \)](#)

Argument	Description
string	The string you want to convert to lowercase letters

Return value: String. Returns [string](#) with uppercase letters changed to lowercase if it succeeds and the empty string ("") if an error occurs.

Upper

Description: Converts all the characters in a string to uppercase.

Syntax: [Upper \(string \)](#)

Argument	Description
String	The string you want to convert to uppercase letters

Return value: String. Returns [string](#) with lowercase letters changed to uppercase if it succeeds and the empty string ("") if an error occurs.

Match

Description: Determines whether a string's value contains a particular pattern of characters.

Syntax: [Match \(string, textpattern \)](#)

Argument	Description
string	The string in which you want to look for a pattern of characters
textpattern	A string whose value is the text pattern

Return value: Boolean. Returns TRUE if [string](#) matches [textpattern](#) and FALSE if it does not. Match also returns FALSE if either argument has not been assigned a value or the pattern is invalid.

Usage: Match enables you to evaluate whether a string contains a general pattern of characters. To find out whether a string contains a specific substring, use the [Pos](#) function.

[Textpattern](#) is similar to a regular expression. It consists of metacharacters, which have special meaning, and ordinary characters, which match themselves. You can specify that the string begin or end with one or more characters from a set, or that it contains any characters except those in a set.

A text pattern consists of [metacharacters](#), which have special meaning in the match string, and **nonmetacharacters**, which match the characters themselves.

See also:

[Metacharacters](#) and sample patterns

Metacharacters

The following tables explain the meaning and use of the metacharacters:

Metacharacter	Meaning	Example
Caret (^)	Matches the beginning of a string	^C matches C at the beginning of a string
Dollar sign (\$)	Matches the end of a string	s\$ matches s at the end of a string
Period (.)	Matches any character	... matches three consecutive characters
Backslash (\)	Removes the following metacharacter's special characteristics so that it matches itself	\\$ matches \$
Character class (a group of characters enclosed in square brackets ([]))	Matches any of the enclosed characters	[AEIOU] matches A, E, I, O, or U You can use hyphens to abbreviate ranges of characters in a character class. For example, [A-Za-z] matches any letter
Complemented character class (first character inside the brackets is a caret)	Matches any character not in the group following the caret	[^0-9] matches any character except a digit, and [^A-Za-z] matches any character except a letter

The metacharacters asterisk (*), plus (+), and question mark (?) are unary operators that are used to specify repetitions in a regular expression:

Metacharacter	Meaning	Example
* (asterisk)	Indicates zero or more occurrences	A* matches zero or more As (no As, A, AA, AAA, and so on)
+ (plus)	Indicates one or more occurrences	A+ matches one A or more than one A (A, AAA, and so on)
? (question mark)	Indicates zero or one occurrence	A? matches an empty string ("") or A

Sample patterns

The following table shows various text patterns and sample text that matches each pattern:

This pattern	Matches
AB	Any string that contains AB; for example, ABA, DEABC, graphAB_one
B*	Any string that contains 0 or more Bs; for example, AC, B, BB, BBB, ABBBC, and so on
AB*C	Any string containing the pattern AC or ABC or ABBC, and so on (0 or more Bs)
AB+C	Any string containing the pattern ABC or ABBC or ABBBC, and so on (1 or more Bs)
ABB*C	Any string containing the pattern ABC or ABBC or ABBBC, and so on (1 B plus 0 or more Bs)
^AB	Any string starting with AB
AB?C	Any string containing the pattern AC or ABC (0 or 1 B)
^[ABC]	Any string starting with A, B, or C
[^ABC]	A string containing any characters other than A, B, or C
^[^abc]	A string that begins with any character except a, b, or c
^[^a-z]\$	Any single-character string that is not a lowercase letter (^ and \$ indicate the beginning and end of the string)
[A-Z]+	Any string with one or more uppercase letters
^[0-9]+\$	Any string consisting only of digits
^[0-9][0-9][0-9]\$	Any string consisting of exactly three digits
^([0-9][0-9][0-9])\$	Any consisting of exactly three digits enclosed in parentheses

See also:

[Match](#) function

Mid

Description: Obtains a specified number of characters from a specified position in a string.

Syntax: `Mid (string, start {, length})`

Argument	Description
string	The string from which you want characters returned
start	A long specifying the position of the first character you want returned. (The position of the first character of the string is 1)
length (optional)	A long whose value is the number of characters you want returned. If you do not enter length or if length is greater than the number of characters to the right of start , Mid returns the remaining characters in the string

Return value: String. Returns characters specified in [length](#) of [string](#) starting at character [start](#). If [start](#) is greater than the number of characters in [string](#), the Mid function returns the empty string (""). If [length](#) is greater than the number of characters remaining after the [start](#) character, Mid returns the remaining characters. The return string is not filled with spaces to make it the specified [length](#).

Pos

Description: Finds one string within another string.

Syntax: `Pos (string1, string2 {, start})`

Argument	Description
string1	The string in which you want to find string2
string2	The string you want to find in string1
start (optional)	A long indicating where the search will begin in string1 . The default is 1

Return value: Long. Returns a long whose value is the starting position of the first occurrence of [string2](#) in [string1](#) after the position specified in [start](#). If [string2](#) is not found in [string1](#) or if [start](#) is not within [string1](#), Pos returns 0.

Usage: The Pos function is case sensitive.

Replace

Description: Replaces a portion of one string with another.

Syntax: `Replace (string1, start, n, string2)`

Argument	Description
string1	The string in which you want to replace characters with string2
start	A long whose value is the number of the first character you want replaced. (The first character in the string is number 1)
n	A long whose value is the number of characters you want to replace

string2 The string that will replace characters in **string1**. The number of characters in **string2** can be greater than, equal to, or fewer than the number of characters you are replacing

Return value: String. Returns the string with the characters replaced if it succeeds and the empty string ("") if it fails.

Usage: If the **start** position is beyond the end of the **string1**, Replace appends **string2** to **string1**. If there are fewer characters after the **start** position than specified in **n**, Replace replaces all the characters to the right of character **start**.

If **n** is zero, then in effect Replace inserts **string2** into **string1**.

Right

Description: Obtains a specified number of characters from the end of a string.

Syntax: `Right (string, n)`

Argument	Description
string	The string containing the characters you want
n	A long specifying the number of characters you want

Return value: String. Returns the rightmost **n** characters in **string** if it succeeds and the empty string ("") if an error occurs. If **n** is greater than or equal to the length of the **string**, Right returns the entire **string**. It does not add spaces to make the return value's length equal to **n**.

Space

Description: Builds a string of the specified length whose value consists of spaces.

Syntax: `Space (n)`

Argument	Description
n	A long whose value is the length of the string you want filled with spaces

Return value: String. Returns a string filled with **n** spaces if it succeeds and the empty string ("") if an error occurs.

String

Description: Formats data as a string according to a specified format mask. You can convert and format date, DateTime, numeric, and time data. You can also apply a format to a string.

Syntax: `String (data {, format})`

Argument	Description
Data	The data you want returned as a string with the specified

formatting. Data can have a date, DateTime, numeric, time, or string data type

format (optional) A [string of the masks](#) you want to use to format the data. The masks consist of formatting information specific to the data type of data. If data is type string, format is required. The format string can consist of more than one mask, depending on the data type of data. Each mask is separated by a semicolon. See Usage for details on each data type

Return value: String. Returns data in the specified [format](#) if it succeeds and the empty string ("") if the data type of data does not match the type of mask specified or [format](#) is not a valid mask.

Usage: For date, DateTime, numeric, and time data, program uses the system's default format for the returned string if you don't specify a format. For numeric data, the default format is the [General] format. For string data, a format mask is required. (Otherwise, the function would have nothing to do.) The format can consist of one or more masks:

- Formats for date, DateTime, string, and time data can include one or two masks. The first mask is the format for the data; the second mask is the format for a null value.
- Formats for numeric data can have up to four masks. A format with a single mask handles both positive and negative data. If there are additional masks, the first mask is for positive values, and the additional masks are for negative, zero, and NULL values.

If the format doesn't match the data type, DB Monitor Expert will try to apply the mask, which can produce unpredictable results.

See also:

[Format symbols](#)

Format Symbols

The following table lists the format symbols that can be used in a format string:

Format Symbol	Description
[General]	Outputs the number in General format.
0	Digit placeholder. If the number contains fewer digits than the format contains placeholders, the number is padded with 0's. If there are more digits to the right of the decimal than there are placeholders, the decimal portion is rounded to the number of places specified by the placeholders. If there are more digits to the left of the decimal than there are placeholders, the extra digits are retained.
#	Digit placeholder. This placeholder functions the same as the 0 placeholder except the number is not padded with 0's if the number contains fewer digits than the format contains placeholders.
?	Digit placeholder. This placeholder functions the same as the 0 placeholder except that spaces are used to pad the digits.
. (period)	Decimal point. Determines how many digits (0's or #'s) are output on either side of the decimal point. If the format contains only #'s left of the decimal point, numbers less than 1 begin with a decimal point. If the format contains 0's left of the decimal point, numbers less than 1 begin with a 0 left of the decimal point.
%	Outputs the number as a percentage. The number is multiplied by 100 and the % character is appended.
, (comma)	Thousands separator. If the format contains commas separated by #'s or 0's, the number is

	output with commas separating thousands. A comma following a placeholder scales the number by a thousand. For Example, the format 0, scales the number by 1000 (e.g., 10,000 would be converted as 10).
E- E+ e- e+	Outputs the number as scientific notation. If the format contains a scientific notation symbol to the left of a 0 or # placeholder, the number is output in scientific notation and an E or an e is added. The number of 0 and # placeholders to the right of the decimal determines the number of digits in the exponent. E- and e- place a minus sign by negative exponents. E+ and e+ place a minus sign by negative exponents and a plus sign by positive exponents.
\$ - + / () : space	Outputs that character. To output a character other than those listed, precede the character with a back slash (\) or enclose the character in double quotation marks (" "). You can also use the slash (/) for fraction formats.
\	Outputs the next character. The backslash is not output. You can also output a character or string of characters by surrounding the characters with double quotation marks (" ").
* (asterisk)	Repeats the next character until the width of the column is filled. You cannot have more than one asterisk in each format section.
[]	Outputs hours greater than 24, or minutes or seconds greater than 60. Place the brackets around the leftmost part of the time code; for Example, [h]:mm:ss would allow the output of hours greater than 24.
"text"	Outputs the text inside the quotation marks.
@	Text placeholder. Text replaces the @ format character.
m	Month number. Outputs the month as digits without leading zeros (e.g., 1-12). Can also represent minutes when used with h or hh formats.
mm	Month number. Outputs the month as digits with leading zeros (e.g., 01-12). Can also represent minutes when used with the h or hh formats.
mmm	Month abbreviation. Outputs the month as an abbreviation (e.g., Jan-Dec).
mmm	Month name. Outputs the month as a full name (e.g., January-December).
d	Day number. Outputs the day as digits with no leading zero (e.g., 1-9).
dd	Day number. Outputs the day as digits with leading zeros (e.g., 01-31).
ddd	Day abbreviation. Outputs the day as an abbreviation (e.g., Sun-Sat).
dddd	Day name. Outputs the day as a full name (e.g., Sunday-Saturday).
yy	Year number. Outputs the year as a two-digit number (e.g., 00-99).
yyyy	Year number. Outputs the year as a four-digit number (e.g., 1900-2078).
h	Hour number. Outputs the hour as a number without leading zeros (1-23). If the format contains one of the AM or PM formats, the hour is based on a 12-hour clock. Otherwise, it is based on a 24-hour clock.
hh	Hour number. Outputs the hour as a number with leading zeros (01-23). If the format contains one of the AM or PM formats, the hour is based on a 12-hour clock. Otherwise, it is based on a 24-hour clock.
m	Minute number. Outputs the minute as a number without leading zeros (0-59). The m format must appear immediately after the h or hh symbol. Otherwise, it is interpreted as a month number.
mm	Minute number. Outputs the minute as a number with leading zeros (00-59). The mm format must appear immediately after the h or hh symbol. Otherwise, it is interpreted as a month number.
s	Second number. Outputs the second as a number without leading zeros (0-59).

ss	Second number. Outputs the second as a number with leading zeros (00-59).
AM/PM am/pm A/P a/p	12-hour time. Outputs time using a 12-hour clock. Outputs AM, am, A, or a for times between midnight and noon; outputs PM, pm, P, or p for times from noon until midnight.

Trim

Description: Removes leading and trailing spaces from a string.

Syntax: `Trim (string)`

Argument	Description
<code>string</code>	The <code>string</code> you want returned with leading and trailing spaces deleted

Return value: String. Returns a copy of `string` with all leading and trailing spaces deleted if it succeeds and the empty string ("") if an error occurs.

WordCap

Description: Sets the first letter of each word in a string to a capital letter and all other letters to lowercase (for example, ROBERT E. LEE would be Robert E. Lee).

Syntax: `WordCap (string)`

Argument	Description
<code>string</code>	A string or expression that evaluates to a string with initial capital letters (for example, Monday Morning)

Return value: String. Returns `string` with the first letter of each word set to uppercase and the remaining letters lowercase if it succeeds and NULL if an error occurs.

CrosstabAvg

Description: Calculates the average of the values returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabAvg can also calculate averages of the expression's values for groups of column values.

Syntax: `CrosstabAvg (n {, column, groupvalue})`

Argument	Description
<code>n</code>	The number of the crosstab-values expression for which you want the average of the returned values. The crosstab expression must be numeric.
<code>column</code>	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog box for which you want intermediate calculations.
<code>groupvalue</code>	A string whose value controls the grouping for the calculation. <code>Groupvalue</code> is usually a

	value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name, as a quoted string.
--	---

Return value: Double. Returns the average of the crosstab values returned by expression **n** for all the **column** values or, optionally, for a subset of **column** values.

Usage: You can use this function only in a crosstab report. This function is meaningful only for the **average** of the **values** of the expression in a row in the crosstab. **NULL** values are ignored and are not included in the average.

CrosstabCount

Description: Counts the number of values returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabCount can also count the number of the expression's values for groups of column values.

Syntax: `CrosstabCount (n {, column, groupvalue})`

Argument	Description
n	The number of the crosstab-values expression for which you want the total number of returned values.
column	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog box for which you want intermediate calculations.
groupvalue	A string whose value controls the grouping for the calculation. Groupvalue is usually a value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name, as a quoted string.

Return value: Long. Returns the number of values returned by expression **n** for all the **column** values or, optionally, for a subset of **column** values.

Usage: You can use this function only in a crosstab report. This function is meaningful only for the **count** of the **values** of the expression in a row in the crosstab. **NULL** values are ignored and are not included in the count.

CrosstabMax

Description: Calculates the maximum value returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabMax can also calculate the maximum of the expression's values for groups of column values.

Syntax: `CrosstabMax (n {, column, groupvalue})`

Argument	Description
n	The number of the crosstab-values expression for which you want the maximum returned value. The expression's data type must be numeric.
column	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog box for which you want intermediate calculations.
groupvalue	A string whose value controls the grouping for the calculation. Groupvalue is usually a value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name, as a quoted string.

Return value: Double. Returns the maximum value returned by expression *n* for all the *column* values or, optionally, for a subset of *column* values.

Usage: You can use this function only in a crosstab report. This function is meaningful only for the *maximum* of the *values* of the expression in a row in the crosstab. *NULL* values are ignored and are not included in the comparison.

CrosstabMin

Description: Calculaes the minimum value returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabMax can also calculate the minimum of the expression's values for groups of column values.

Syntax: `CrosstabMin (n {, column, groupvalue})`

Argument	Description
<i>n</i>	The number of the crosstab-values expression for which you want the minimum returned value. The expression's data type must be numeric.
<i>Column</i>	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog box for which you want intermediate calculations.
<i>Groupvalue</i>	A string whose value controls the grouping for the calculation. <i>Groupvalue</i> is usually a value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name, as a quoted string.

Return value: Double. Returns the minimum value returned by expression *n* for all the *column* values or, optionally, for a subset of *column* values.

Usage: You can use this function only in a crosstab report. This function is meaningful only for the *minimum* of the *values* of the expression in a row in the crosstab. *NULL* values are ignored and are not included in the comparison.

CrosstabSum

Description: Calculates the sum of the values returned by an expression in the values list of the crosstab. When the crosstab definition has more than one column, CrosstabSum can also calculate the sum of the expression's values for groups of column values.

Syntax: `CrosstabSum (n {, column, groupvalue})`

Argument	Description
<i>n</i>	The number of the crosstab-values expression for which you want the sum of the returned values. The expression's data type must be numeric.
<i>Column</i>	The number of the crosstab column as it is listed in the Columns box of the Crosstab Definition dialog box for which you want intermediate calculations.
<i>Groupvalue</i>	A string whose value controls the grouping for the calculation. <i>Groupvalue</i> is usually a value from another column in the crosstab. To specify the current column value in a dynamic crosstab, rather than a specific value, specify @ plus the column name, as a quoted string.

Return value: Double. Returns the total of values returned by expression `n` for all the `column` values or, optionally, for a subset of `column` values.

Usage: You can use this function only in a crosstab report. This function is meaningful only for the `sum` of the `values` of the expression in a row in the crosstab. `NULL` values are ignored and are not included in the `sum`.

Operators in Sort and Filter Expressions

An operator is a symbol or word in an expression that performs an arithmetic calculation or logical operation; compares numbers, text, or values; or manipulates text strings.

The database buffer supports the following types of operators:

- Arithmetic for numeric data types
- Relational for all data types
- Logical for all data types
- Concatenation for string data types

Arithmetic operators

These are the arithmetic operators:

Operator	Meaning	Example
+	Addition	SubTotal + Tax
-	Subtraction	Price - Discount
*	Multiplication	Quantity * Price
/	Division	Discount / Price
^	Exponentiation	Rating ^ 2.5

Relational and logical operators

Logical operators can join relational expressions to form more complex boolean expressions.

The result of evaluating a boolean expression is always TRUE or FALSE.

These are the relational and logical operators:

Operator	Meaning	Example
=	Equals	Price = 100
>	Greater than	Price > 100
<	Less than	Price < 100
<>	Not equal	Price <> 100
>=	Greater than or equal	Price >= 100
<=	Less than or equal	Price <= 100
NOT	Logical negation	NOT Price = 100
AND	Logical and	Tax > 3 AND Ship < 5
OR	Logical or	Tax >3 OR Ship < 5
IN	SQL SELECT IN condition	Status IN ('A', 'S', 'R')

BETWEEN	SQL SELECT BETWEEN condition	Ship BETWEEN 3 AND 5
LIKE	SQL SELECT LIKE condition	Brand LIKE 'Micro%'

Do not use **IS NULL** and **NOT IS NULL** for NULL value checking, use embedded [isNull\(\)](#) function instead.

Concatenation operator

The concatenation operator joins the contents of two string expressions to form a longer value. You can concatenate strings only.

Operator	Meaning	Example
+	Concatenate	"cat " + "dog"

Operator precedence in expressions

To ensure predictable results, all operators in an expression are evaluated in a specific order of precedence. When the operators have the same precedence, the database buffer evaluates them left to right.

The following table lists the operators in descending order of precedence:

Operator	Purpose
()	Grouping
^	Exponentiation
*, /	Multiplication and division
+, -	Addition and subtraction; string concatenation
IN, LIKE, BETWEEN	SQL SELECT statement conditions
NOT	Logical negation
=, >, <, <=, >=, <>	Relational operators
AND, OR	Logical AND and logical OR

Hot Keys and Shortcuts

Key	Action
CTRL F4	Close active sheet
ALT F4	Close active dialog box, if no dialog active -close the application
CTRL N	Insert new row; Start new SQL file in DBA Notepad
CTRL D	Delete current row; Delete highlighted rows on Flow Dataview tab; Kill session in Connection Manager; Drop active object on Table design and relationships window
CTRL O	Open file in DBA Notepad; Start analyzing table(s) process in Table Analyzer; Start analyzing index(es) process in Index Analyzer, Start analyzing cluster(s) process in Cluster Analyzer
CTRL SHIFT O	Import code of stored procedure, function, trigger, or package into DBA Notepad
CTRL S	Save changes
CTRL P	Print active sheet
CTRL R	Execute statements in DBA Notepad
CTRL X	Cut selection
CTRL C	Copy selection

CTRL V	Paste text from clipboard into current column
CTRL A	Select all
CTRL Z	Undo changes
CTRL Y	Redo changes undid before
CTRL F	Start search
F3	Repeat previous search in DBA Notepad
CTRL H	Replace text in DBA Notepad
CTRL G	Go to row on report/datasheet; Go to line in DBA Notepad
CTRL E	Start DBA Notepad
F4	Open Filter Dialog
F5	Open Sort Dialog
F7	Start Database Object Navigator
CTRL +	Select all items (rows) in select table/index/schema/etc dialog boxes; Select all rows in Table Analyzer; Index Analyzer; Cluster Analyzer for batch processing
CTRL -	Deselect all selected items (rows) in select table/index/schema/etc dialog boxes; Deselect all selected rows in Table Analyzer; Index Analyzer; Cluster Analyzer for batch processing
CTRL T	Start table/view design and data manipulation
CTRL K	Start Security Navigator
TAB	Move to next field on datasheet
SHIFT TAB	Move to previous field on datasheet
CTRL TAB	Switch to next open sheet
ALT TAB	Switch to another Windows application or to another DB Monitor Expert instance
ALT SPACE	Activate system menu for active window or object
ESC	Close active dialog (equivalent to Cancel button); Undo last changes in the field that is being edited on result set datasheet
F1	Get online help

SQL Editor and DBA Notepad

Key	Action
CTRL + Left	one word left
CTRL + Right	one word right
CTRL + Up	to the beginning of paragraph
CTRL + Down	to the end of paragraph
Home	to the beginning of the line
End	to the end of the line
PageDown	page down
PageUp	page up
CTRL + Home	to the beginning of the text
CTRL + End	to the end of the text
CTRL + PageUp	to the beginning of the visible text
CTRL + PageDown	to the end of the visible text
SHIFT	when combined with all previous keys (including mouse) expands current selection
Insert	changes write mode
Delete	deletes one symbol (or clear selection)
BackSpace	deletes one symbol (or clear selection)
CTRL + R	one word to right
CTRL + L	one word to left
CTRL + U	to the beginning of the visible text
CTRL + D	to the end of the visible text
CTRL + Z	Undo
CTRL + Y	Redo
CTRL + X	Cut
CTRL + C	Copy
CTRL + V	Paste
CTRL + A	Select All

CTRL + F	Find
CTRL + H	Replace
CTRL + G	Go to Line Number
CTRL + Ins	Copy
SHIFT + Ins	Paste
SHIFT + CTRL + Ins	Duplicate
CTRL + Del	Clear
SHIFT + Del	Cut
CTRL + BackSpace	Undo
SHIFT + CTRL + BackSpace	Redo
F9	Cancel Retrieval Operation
CTRL F4	Close active sheet

Installation and Uninstallation

Installation

The DB Tools Setup program provides very straightforward interface for DB Tools installation. Simply follow the Installation Wizard instruction that will guide you through the entire installation process.

If you are going to use an ODBC interface for the database connection, configure required data sources using the ODBC Administrator that can be found in the Windows Control Panel.

See also:

[System Requirements](#)

Uninstallation

Front-end:

The DB Tools supports standard uninstallation mechanism for removing program files from the computer. To uninstall the DB Tools do the following:

- 1 Click Windows **Start** button, from the Start Menu select **Settings**, then **Control Panel**.
- 2 Double-click **Add/Remove Programs**.
- 3 Select the **DB Tools** item in the programs list, click **Add/Remove** button
- 4 Delete all left files from the DB Tools home directory. Delete the home directory.

Back-end:

- 1 Execute UNINSTALL.SQL script using DB Tools SQL Editor, SQL*Plus or any other tool of your choice.

System Requirements

Minimum Hardware Requirements

- 1 Intel-based machine running one of the following operating system:
 - Windows NT 4.0 (server or workstation)
 - Windows 2000 (server or workstation)
 - Windows XP (server or workstation)
 - Windows 95
 - Windows 98
 - Windows Me
- 2 At least 128 MB RAM
- 3 35 MB disk space for full installation
- 4 VGA monitor or better

Recommended Configuration

- 1 Pentium class CPU 600 MHz or better

- 2 256 MB RAM or better
- 3 SVGA 256-color or better monitor

Software Requirements

Server:

- 1 Oracle Server version 7.3 and above

Client:

- 1 Windows 95, Windows 98, Windows Me, Windows 2000, Windows NT 4.0 or Windows XP
- 2 Required database client software: SQL*Net client software (Version 2 or 8 or 9 or 10); Required Support Files (RSF)
- 3 ODBC (optional)
- 4 Remote Procedure Call (RPC) Service (installed by default)

The Dynamic Performance Tables

The Oracle Server contains a set of underlying "tables" that are maintained by the Server and accessible to the Database Administrator (DBA) user SYS. These tables are called dynamic performance tables because they are continuously updated while a database is open and in use, and their contents relate primarily to performance. **After installation, only "username" SYS has access to the dynamic performance tables. However, access to these tables is required for any user who wants to run the DB Tools (all).** The UTLMONTR.SQL script can be run to grant access to PUBLIC on all of the dynamic performance tables. See your installation or user's guide for details on running UTLMONTR.SQL.

SQL Debugger: DBMS_DEBUG package must be installed in the destination database

SQL Profiler: DBMS_PROFILER package must be installed in the destination database

Java Editor (Java code compilation options): Java Run-time must be installed and configured in the database.



Note: In order to run DB Tools without problems you should have SELECT_CATALOG_ROLE and EXECUTE_CATALOG_ROLE roles granted to you. Having SELECT ANY TABLE privilege is highly recommended but not required. For certain types of operation supported by DB Tools you may need other privileges such as AUDIT ANY, CREATE ANY TRIGGER, ANALYZE ANY, EXECUTE ANY, ADMINISTER RESOURCE MANAGER and some other DBA-level privileges.

Portability

Oracle database software is ported to work under different operating systems and is the same on all systems. The DB Tools can work with any Oracle database (version 7.3 and above) running on any operating system.

See also:

[Installation and Uninstallation](#)

Connecting to Oracle databases

Required Oracle Client Software

What is SQL*Net?

SQL*Net is the Oracle product that provides connectivity between clients and Oracle servers. This product was renamed to Net8 in the Oracle8 release, Net9 in the Oracle 9i release and then to Net10 in Oracle 10g release. When this document refers to SQL*Net, it means SQL*Net for Oracle7 and Net for Oracle8 and later.

You must install Oracle client software SQL*Net before attempting to connect using DB Tools.

When you install SQL*Net, be sure you install the product specific to your operation system. Remember that Oracle distinguishes between Windows 95/98 and Windows NT/2000/XP/Vista.

Important: Make sure your Oracle client software home directory is added to the system PATH environment variable

What you must install?

SQL*Net

DB Tools and Oracle will need this for connections.

Network Adapter Layer

The Oracle Network Adapter handles communication between the Oracle Call Interface and you network protocol. For example, if you communicate with your Oracle server using the TCP/IP, you must install the Oracle TCP/IP protocol adapter. If you use different protocols at different times, you can install more than one protocol adapter.

Testing Connection Outside of DB Tools

Before attempting to connect from DB Tools, you should verify that you can connect from SQL*Plus. If you cannot connect from SQL*Plus, you have no hope of connecting through DB Tools.

How DB Tools Talks to Oracle

Overview

DB Tools communicates with Oracle through the Oracle Call Interface (OCI). This interface is not used by SQL*Plus. You must be careful as to what conclusions you can draw when something works in SQL*Plus but not in DB Tools.

DB Tools determines the network interface DLL based on the first three characters of your DBMS specification.

* Oracle 7 interfaces are also supported with Required Support Files (RSF) included with Net8

Connect to Database Dialog

The same dialog box for establishing new database connections is used in all DB Tools components. The dialog can be displayed as a separate window or be embedded as page in a graphical multi-step wizard dialog.

For more information about the Connect dialog, available controls and connection options see [Connect To Database Dialog](#) topic in the Common Dialogs and Controls Chapter.

Common Problems

Missing network module

If you select a DBMS driver for an interface that you have not installed, you will get a connect error indicating that your DBMS is not supported.

Path errors

When DB Tools connects to Oracle, the DB Tools module attempts to load the corresponding OCI module. This load is no different from any other load in that the module must be in your <windows>\system directory or in a directory on your path. If the path does not include your <Oracle home>/bin directory, the DLL will not be found and an error will occur during connect. The error will indicate that your DBMS is not supported.

Missing Required Support Files

In order to connect from DB Tools, you must have installed the Oracle Required Support Files that correspond to your DBMS value. For example, if you specify O73 as your DBMS driver value, you must have installed Required Support Files version 7.3.x.

Improperly Installed Required Support Files

Required Support Files should be installed using the Oracle Installer. If you copy files from one location or system to another, you may run into problems. If you install SQL*Net on a Local Area Network where it will be accessed by multiple users, make sure you have followed procedures provided by Oracle for Network Installation. It is not as easy as installing from a single client and letting everyone access the resulting files. If you have installed Net8 and then install another product that includes Oracle Required Support Files, you may encounter a problem loading OCIW32.DLL. To remedy the situation, re-install the 8.0.x Required Support Files.

Incorrect server parameter in transaction Object

Because of changes in the Oracle OCI interface, the format of server parameter may differ between various releases of SQL*Net. If the Oracle service name as specified in your TNSNAMES.ORA file was orasrvr1.world, the server parameter for a DB Tools connection is shown in the following table:

DBMS	Server Value
O73	@orasrvr1.world
OR8 and later	orasrvr1.world

Debugging

When DB Tools connects to Oracle, DB Tools simply calls a function in the Oracle OCI layer. If you are having a problem connecting, it may be useful to run a SQL*Net "Admin" level trace. This trace will show much of the SQL*Net activity during the failed connects. There may be Oracle error messages that are more useful than the message echoed back to DB Tools.

Special Considerations with Oracle 7.3

In general, SQL*Net 2.x is backward compatible. For example, If you upgraded your Server to SQL*Net 2.2 and still have SQL*Net 2.1 on your client, you will still be able to communicate. There is, however, one exception. If you are running SQL*Net 2.3 on the server, you should also run 2.3 or above on the client. There was a problem with the 2.2 SQL*Net software that was discovered after SQL*Net 2.3 was released. Rather than going back and fixing 2.2, they suggest an upgrade to 2.3. The problem will not prevent a connection. It will, however, cause other connection-related problems while running DB Tools.

Special Considerations with IPX / SPX

The following table shows the versions of SQL*Net required on 32 bit platforms. Netware Client32 is the only supported IPX stack. NWLink from Microsoft is not supported.

Platform	SQL Net Version
Windows 95/98	SQL*Net 2.3.2.1.4 or higher
NT 4.0 and later	SQL*Net 2.2 or higher

Special Considerations for NT 4.0 or above

You must be running SQL*Net 2.3.2.1.4 or above.

Common Error Messages

DBMS <DBMS name> not supported in your current installation

DB Tools native driver DLL is missing or not in path.

Oracle DLL loaded from PBxxx060.DLL missing or not in path.

Oracle Required Support Files needed by DB Tools DLL are not installed.

Dynamic Link Library <dll name> could not be found in the specified path

DLL missing

Directory containing the DLL not in path

If the DLL is ORA803.DLL, check to see if the file is on your system. If it is, it should be on your path. If the file is not there, either you have not installed Net8 or when you installed Net8, you did not install the Oracle 8.0.x Required Support Files.

Error while trying to retrieve text for error ORA_nnnnn

Occurs during connection if there is no Oracle registry entry in HKEY_LOCAL_MACHINE\SOFTWARE

Occurs when the NET20 registry entry in HKEY_LOCAL_MACHINE\SOFTWARE\Oracle is incorrect.

Initialization of the dynamic link library <Oracle_home\bin\OCIW32.DLL path> failed. The process is terminating abnormally.

Oracle DLL ORA73.DLL does not exist or is not in path.

ORA-03121: No interface driver connected - function not performed

(DBMS O73) The server name in the database profile was not in the format @<service name> or the SQL*Net software was not installed properly. Probably missing the "@" sign preceding the service name.

ORA-12154: TNS:could not resolve service name

(DBMS O73) The tnsnames.ora file was not found in Oracle_HOME\NETWORK\ADMIN directory.

(DBMS OR8) The tnsnames.ora file was not found in Oracle_HOME\NET80\ADMIN directory.

(Oracle 7.x) No matching service name in tnsnames.ora file. Be sure to include the complete name (including the domain) as it appears in the tnsnames.ora file. For example, if the entry is orsrvr.world in the tnsnames.ora entry, the server specified in the database profile would be @orsrvr.world. This is true even if the service name specified in SQL*Plus was orsrvr.

(DBMS OR8) If the service name contains a preceding "@" or "@TNS:", this error will occur. If the specified service name was not found in the tnsnames.ora, this error will occur.

ORA-12203: TNS:Unable to connect to destination

The tnsnames.ora file does not exist or cannot be located by the Oracle DLLs.

The network adapter layer was not installed.

ORA-12224 TNS: No listener

(DBMS OR8) The host pointed to by the host entry in your tnsnames.ora file did not have a listener running on the specified port. The host, the port or both were incorrectly specified or the listener has stopped running on the server.

ORA-12505 TNS: Listener could not resolve SID given in connect descriptor

(DBMS OR8) The server and listener were found, but the SID in the tnsnames.ora file was incorrect.

ORA-12545: Connect failed because target host or object does not exist

(DBMS OR8) The host entry in your tnsnames.ora file points to a non-existent server.

Unable to load the requested database interface

Make sure that the DB Tools network module associated with your specified DBMS is installed and on your path.

Connecting to Oracle using Oracle ODBC Drivers

Configure the ODBC Data Source

You can create a new data source by running the ODBC administrator. The **Connect String** in the data source is the service name as found in your tnsnames.ora file.

Note for Personal Oracle: The connection to Personal Oracle bypasses SQL*Net. The SQL*Net Connect string is "2:<SID>" where <SID> is the instance ID. The default SID is "ORCL".

Refer to the Oracle On-Line help for assistance when setting up the data source.

Verify Your Oracle ODBC Connection

Connecting to your database from SQL*Plus verifies that your Oracle client software was installed and configured correctly. If there is a problem with the installation or configuration of your Oracle client software, you will not be able to connect using DB Tools. If you can connect using SQL*Plus, then attempt to connect with another ODBC tool using the data source you created. Oracle ships an ODBC utility with SQL*Net 2.3.x called Oracle ODBC Test. You can use this utility to test the data source connection. Some other common tools are MS Query and MS test. Connecting with another tool makes sure that ODBC is correctly installed and the data source is configured properly. You should verify your ODBC connection outside of DB Tools before calling Technical support for assistance.

Common Connect Errors

DBMS ODBC not supported in your current installation

ODBC32.DLL not found (Windows 95). PBODB60.DLL not found in path.

IM002 - Data Source name not found and no default driver specified

The data source in the connect string is not valid.

IM003 - Specified driver could not be loaded

The driver specified in the ODBC data source could not be loaded. This happens if the driver was moved or deleted after the data source was created. This error will also occur if the Required Support Files are not installed or are not in the path.

ORA-12154:TNS:could not resolve service name

The tnsnames.ora file was not found in Oracle_HOME\NETWORK\ADMIN directory. This error also occurs if the service name (either from the server in the data source or the DBQ value in the connect string) is not found in the tnsnames.ora file. Remember that there should be no "@" or "@tns:" in this value.

File Error - Cannot find ODBC32.DLL

ODBC32.DLL missing or not in path

Creating ODBC data source for Oracle database

To add a data source:

1. From the installed ODBC drivers list, select the name of the Oracle driver that the data source will use.
2. Choose the **OK** button.
3. The Oracle ODBC Driver dialog box is displayed. Enter information about the data source:

Field	Meaning
Data Source Name	A name by which you will identify the data source. For example, "Personnel Data"
Description	A description of the data in the data source. For example, "Hire date, salary history, and current review of all employees."
SQL*Net Connect String	An Oracle connect string that specifies the location of the copy of Oracle RDBMS from which the driver will retrieve data.

4. Choose the **OK** button.

DB Monitor Expert

Overview

As Oracle customers migrate from large, centralized mainframe systems to smaller distributed systems, administration tasks become increasingly difficult. The Oracle database administrator (DBA) is often responsible for managing a highly diverse combination of local and remote systems and discovering new and more complex administration issues.

Databases are also getting much larger, more complex, and more in need of tuning, creating the need for powerful management and monitoring tools capable of performing many tasks in real-time. Large, mission critical databases need highly effective backup tools, good performance tuning tools and a host of online diagnostic tools.

In order to properly manage such mission critical databases the DBAs must have sufficient information about their database. Only regular database monitoring allows DBA gather enough information about database, identify technical problems in early stage and finally make the database easy to maintain, not fragmented, fast, cost-effective in terms of reducing downtime and avoiding user frustrations.

DB Monitor Expert and other components included in the DB Tools package provide Oracle DBAs and application developers with a powerful set of sophisticated tools for monitoring, tuning, exploring, managing, auditing, coding and benchmarking Oracle databases. By using these tools, Oracle DBAs and application developers can make significant gains in productivity and manageability.

Why you have to monitor your database regularly

As Oracle customers migrate from large, centralized mainframe systems to smaller distributed systems, administration tasks become increasingly difficult. The Oracle database administrator (DBA) is often responsible for managing a highly diverse combination of local and remote systems and discovering new and more complex administration issues.

Databases are also getting much larger, more complex, and more in need of tuning, creating the need for powerful management and monitoring tools capable of performing many tasks in real-time. Large, mission critical databases need highly effective backup tools, good performance tuning tools and a host of online diagnostic tools.

In order to properly manage such mission critical databases DBA must have sufficient information about them. Only regular database monitoring allows DBA gather enough information about database, identify technical problems in early stage and finally make the database easy to maintain, not fragmented, fast, cost-effective in terms of reducing downtime and avoiding user frustrations.

DB Monitor Expert provides DBA with easy-to-use effective tools that help monitor and administer distributed Oracle databases from a single console.

In addition, DB Monitor Expert provides the application developers with easy-to-use effective tools that help tune application SQL queries. According to Oracle: 70 to 80% of all Oracle server performance problems are remedied by tuning the SQL statements.

See also:



- [Oracle database architecture](#)
- [Oracle database memory structures and processes](#)
- [Oracle database security](#)

Active Desktop

The Active Desktop allows you to create customized graphical views of your Oracle databases. You can create and modify graphical presentations of the databases. Each of these presentations consists of two elements: a database icon and a label combining database location, Oracle server name and the database name.

When you double-click on the database icon, DB Monitor Expert connects to the activated database. Use dragging to move already created presentations to required location. On right mouse click, DB Monitor Expert displays context-sensitive pop-up menu. Use that menu to create new database presentations, modify or delete existing ones. The Active Desktop can be hidden or shown by selecting correspondingly **Show Desktop/Hide Desktop** command from the menu.

You can change default bitmap used as a background for the Active Desktop. See [Options](#) topic for details. For example, you can use a map presenting your city or state as the Active Desktop background picture. For example, the DBA who is responsible for the databases in New York offices can attach a New York City map. The developer who is working with several Oracle database department servers running on different floors can create a view of the company building.

You can determine at a glance the connection status of a database on the Active Desktop. An icon representing the database that DB Monitor Expert is currently connected to has a small green traffic light displayed next to it . All other database icons have red light next to them .

If you want to save Active Desktop configuration so that it will be available next time you run DB Monitor Expert, use right mouse click anywhere on the Active Desktop. A popup menu will appear. Choose **Save Desktop** command from the menu.

If you do not want to use the Active Desktop, you can disable it in the Program [Options](#).

Performance Graphs (Graphs menu)

DB Monitor Expert provides quick comprehension of system status by presenting information graphically. All DB Monitor Expert graphs found under **Graph** menu are recent-past and real-time graphs. They represent different views of data you get using capabilities of real-time monitors found under **Monitor** menu. Most graphs show "short-term" historical statistic, which you may use for in-depth analysis of database applications activity. For all real-time graphs you can customize [refresh rate](#) and number of checkpoints for the graph. Note that total time interval covered by the graph is *number of checkpoints* times *refresh rate* for that graph. To customize refresh rate and number of checkpoint, click the Options button on the [toolbar](#)



. An [Option dialog](#) box will appear.

Note that the number of checkpoints has no effect for **Database Actions**, **SQL Activity** graphs and some other graph types. You can also right-click any graph to get some additional features accessible from a popup menu that appears in response to the click.

Performance Summary Graphs

Real-time Statistics Graphs

This graph displays this graph displays comprehensive real-time summaries of various aspects of your database performance on a single screen, including Buffer Cache Hit, Dictionary Cache Hit, Shared Pool Hit, Memory Sorts Hit, Recursive Calls Ratio, Non-index Lookup Ratio, User activity, Number of active Cursors and Locks, File I/O Throughout, and Rollback Segment I/O Throughout

Recent Statistics Graphs

This graph displays summaries of various aspects of your database performance as a function of time. By default all line graphs are drawn for the last 10 hours. You can customize that interval using the "graph scale" control displayed on the top of the graph window. Recent Statistics Graphs include Number of Active Sessions, Buffer Cache Misses, CPU Usage, Memory Usage, and File I/O Throughout. The information displayed by this monitor is collected by the [Background Database Performance Statistics Collection](#) procedure, which must be installed in the database in order to run Recent Statistics Graphs.

Historical Statistics Graphs

This graph displays summaries of various aspects of your database performance as a function of time. By default all line graphs are drawn for the last 90 days. You can customize that interval using the "graph scale" control displayed on the top of the graph window. Historical Statistics Graphs include Number of Active Sessions, Buffer Cache Misses, CPU Usage, Memory Usage, and File I/O Throughout. The information displayed by this monitor is collected by the [Background Database Performance Statistics Collection](#) procedure, which must be installed in the database in order to run Historical Statistics Graphs.

Database Actions Graph

This graph displays summary statistic for database activity. It shows percent of different types of SQL statements executed by the database.

SQL Activity Graph

This graph displays SQL Area usage and helps to pinpoint not tuned database applications that send offensive SQL statements to the database. Such SQL statements have hit ratio (buffer gets vs. disk reads) significantly above average.

SGA Memory Usage Graph

This graph displays SGA memory distribution between database buffers, redo buffers, fixed memory size, and variable memory size. Both Percents and Mbytes are displayed. The graph also shows variable memory distribution between shared pool, java pool, large pool and other objects.

License/Users Logged On Graph

This graph displays number of active database sessions as a function of time. By default the horizontal dotted green line shows high water mark for the number of sessions since this graphical monitor started. Use this graph to monitor whether the number of database sessions violates your seat-based license and if not how close you are. You can also use to monitor picks of user activity.

Rollback Segment (Gets/Writes) Graph

This graph displays disk gets and disk writes for all rollback segments; helps to examine work load for rollback segments.

Rollback Segment (Hits) Graph

This graph displays disk gets and disk writes performance for all rollback segments; helps to examine performance for rollback segments.

Rollback Segment Transactions Graph

This graph displays number of active database transactions by rollback segment. This is also a good indicator of the overall database activity.

Redo Log Latch (Hits) Graph

This graph displays gets and immediate gets for different types of redo log latches; helps to examine performance for redo logs and buffers.

Redo Log Latch (Misses) Graph

This graph displays misses and immediate misses for different types of redo log latches; helps to examine performance for redo logs and buffers.

CPU Usage Graph

This graph displays total CPU time allocated to all user sessions as a function of time

Memory Usage Graph

This graph displays total memory size in Mbytes allocated for all user sessions as a function of time

Buffer Cache Graph

This graph displays logical and physical reads performance; helps to find not tuned database applications that cause decreasing effectiveness of the cache.

Dictionary Cache Graph

This graph displays changes of dictionary hit ratio; helps to find database applications that need bigger dictionary cache.

Recursive Calls Graph

This graph displays frequency of recursive calls; helps to identify not tuned SQL statements and database applications.

Sorts Graph

This graph displays performance of the database sort operations.

Waits Graph

This graph displays various database waits. Run this graph together with the **Rollback Segment** graphs and/or **Redo Latch** graphs to detect whether rollback segments or redo logs cause any waits.

File Input/Output (Hits) Graph



This graph displays reads and writes performance for all database files; helps to resolve I/O bottlenecks.

File Input/Output (Blocks) Graph

This graph displays block reads and writes performance for all database files; helps to resolve I/O bottlenecks and balance distribution of I/O.

Custom Graph

In addition to a variety of predefined recent-past graphs available in DB Monitor Expert, you can also define your own graphs. User-defined graphs can address your database specific requirements. Using custom graphs, you can also automate problem detection and correction, and optionally execute a job to fix the problem and generate alarm message that may be sent via E-mail, pager or beeper to you or other people. To create a new graph:

1. Select **Graph/Custom Graph** menu. DB Monitor Expert will open the **Custom Graph** dialog. The dialog window will display all already created custom graphs.
2. Click the **New** button.
3. Enter valid name in the **Name** field on the **Properties** tab. Note that name may consist spaces.
4. Specify appropriate timer interval in the **Interval** field. The number must be in range from 1 to 32767 seconds.
5. Specify the number of graphed data-points in the **Checkpoints** field. The number must be in range from 3 to 99.
6. Type valid SQL SELECT statement in the **Select** field. Alternatively, you can paste SQL from the clipboard. To do this, use keyboard shortcut CTRL V or use right mouse click on the **Select** field then from popup menu choose **Paste** command. Also, you can click on the **Open File**  button to paste SQL from the file. This SQL statement will be used to gather statistics to be displayed in the monitor you are defining. Make sure that the SQL statement does not end with a semicolon (;). To avoid unclear column names in the graph syntax always use aliases in the select statement (see [examples](#)).
7. Click on the **Preview** tab to customize display properties of the graph.
8. Select column from the **Value** drop-down list. That column will be graphed. Note that you can pick column hits from the **Value** list then instead of column's absolute value DB Monitor Expert will graph changes per second for that column.
9. If you want to display multiple graphs (series of characteristics) simultaneously, use the **Series** drop-down list to select column that will be used for series. For display of single value leave **Series** field blank or select **<None>** from the list.
10. Select required graph style.
11. Click on the **Threshold** tab to customize new monitor behavior. If required, enter threshold condition that the monitor will verify every time a new sampling will occur.
12. If you wish, you can specify *fixit* job, which the monitor performs when the threshold condition is detected. In the **SQL File** field type SQL file name where *fixit* job specified. Alternatively, you can click **Open File**  button to browse and paste the SQL file name. For example, you can have the monitor for monitoring the tablespace TEMP in the database SALES and generate an alert if the tablespace runs out of free space. You can also have a data-file automatically added to the tablespace if this event occurs.
13. To generate alert message, fill all fields in the **Action** box. Following methods are presently supported: [E-mail](#), [Pager](#), and [Beeper](#). Note that the Beeper method is provided so that you can send messages to "Numeric Only" pagers. It does not use the TAP protocol. DB Tools simply takes the phone line off-hook, dials the beeper or pager phone number, waits x number of seconds, touch tones the digits to be displayed on the pager and hangs up.
14. Click the **Save** button to save the new graph definition. The graph can be used again later on.
15. Click the **Run** button to start the graph you have defined. Note that you can run the graph without doing save first, but by default DB Monitor Expert does not save the graph definition for you until you click the **Save** button. So that if you do not save the definition you will not be able to reuse this graph later.
16. Now you can build another graph or click the **Exit** button to close Custom Graph dialog.

The following 3 examples of user-defined graph are included in the standard installation package:

1. Active Sessions

```
select count(distinct sid) active_sessions
from v$session
where type = 'USER'
```

2. Executions By Segment Type

```
select sum(executions) executions,
type segment_type
from v$db_object_cache
where type not in ('NON-EXISTENT', 'NOT LOADED')
group by type
```

3. SQL*Net Hits

```
select sum(v.value) bytes_sent_received,
n.name
from v$statname n,
v$sesstat v
where v.statistic# = n.statistic# and
n.name in ('bytes sent via SQL*Net to client',
'bytes received via SQL*Net from client')
group by n.name
```

See also:

[Monitor connections and resources](#)
[Historical trends](#)
[Reports](#)
[Data manipulation and data export](#)
[Customization](#)

Exporting Graphs

Use **File/Export** menu in DB Monitor Expert to save screenshots of open graphical monitors to various file formats. Saved files can be then archived or emailed as any other file type. You can use your favorite image editor or web browser to open previously saved graphic files for viewing.

DB Tools currently supports JPG, BMP and WMF file formats.

Background Database Performance Statistics Collection Job

DB Tools allows you to setup an automated procedure for periodically capturing various database dynamic performance statistics and saving them in ORA_MONITOR.DB_WATCH table. Captured statistics can provide invaluable help in troubleshooting database performance issues, monitoring service availability for mission critical applications as well as it can aid in database capacity planning and analysis. All collected statistics are consolidated and saved in a single table as historical database and application performance snapshots. This information can be easily queried, analyzed and reported using any tool of your choice. Several DB tools reports also use the collected data.

The **Background Database Statistics Collection** job can be installed using DB Tools [Database Setup utility](#).

By default the installed job will run every 15 minutes. As the job will run it will automatically save average numbers for every hour creating in total 24 records every day (which is just 8760 records a year).

Captured statistics include all major database performance metrics as well as statistics for top 5 wait events, top 5 active users and top 5 applications causing waits and affecting the overall database performance. The following table describes columns of the ORA_MONITOR.DB_WATCH table.

Column Name	Data Type	Description
TIMESTAMP	DATE	Time when statistics were collected or consolidated for the past hour
STAT_TYPE	CHAR(1)	The value in this column could be either C or H. <ul style="list-style-type: none"> • C – means CURRENT – represent statistics for the current hour • H – means HOURLY – represent past hourly statistics
BUFFER_MISSES	NUMBER	Percent of Db buffer misses (sum of values in BUFFER_MISSES and BUFFER_HITS columns should be always equal to 1)
BUFFER_HITS	NUMBER	Percent of Db Buffer hits (sum of values in BUFFER_MISSES and BUFFER_HITS columns should be always equal to 1)
DICT_MISSES	NUMBER	Percent of Dictionary Cache misses (sum of values in DICT_MISSES and DICT_HITS columns should be always equal to 1)
DICT_HITS	NUMBER	Percent Dictionary Cache hits (sum of values in DICT_MISSES and DICT_HITS columns should be always equal to 1)
LIBRARY_MISSES	NUMBER	Percent of Library Cache misses (sum of values in LIBRARY_MISSES and LIBRARY_HITS columns should be always equal to 1)
LIBRARY_HITS	NUMBER	Percent of Library Cache hits (sum of values in LIBRARY_MISSES and LIBRARY_HITS columns should be always equal to 1)
SORT_MISSES	NUMBER	Percent of in memory sorts misses (sorts requiring disk I/O) (sum of values in SORT_MISSES and SORT_HITS columns should be always equal to 1)
SORT_HITS	NUMBER	Percent of in memory sorts hits (sorts not requiring disk I/O) (sum of values in SORT_MISSES and SORT_HITS columns should be always equal to 1)
RECURSIVE_CALLS	NUMBER	Percent of recursive calls (sum of values in RECURSIVE_CALLS and OTHER_CALLS columns should be always equal to 1)
OTHER_CALLS	NUMBER	Percent of non-recursive calls (sum of values in RECURSIVE_CALLS and OTHER_CALLS columns should be always equal to 1)
LONG_SCANS	NUMBER	Percent of long table scans (sum of values LONG_SCANS and OTHER_SCANS should be always equal to 1)
OTHER_SCANS	NUMBER	Percent of short table scans (sum of values LONG_SCANS and OTHER_SCANS columns should be always equal to 1)
FILE_READS	NUMBER	Total physically file reads in Kbytes performed since the database instance startup
FILE_WRITES	NUMBER	Total physically file writes in Kbytes performed since the database instance startup
FILE_THROUGHOUT	NUMBER	I/O throughout Kbytes/sec. This value is calculated as the total number of Kbytes red/written from/to all database files between 2 statistics collection job runs divided by the number of seconds between job runs.
ROLLBACK_GETS	NUMBER	Total number of rollback gets since the database

		instance startup
ROLLBACK_WRITES	NUMBER	Total number of rollback writes since the database instance startup
USER_SESSIONS	NUMBER	Maximum number of user sessions logged on (sum of values in SES_IDLE and SES_ACTIVE columns should always match the value in the USER_SESSIONS column)
SES_IDLE	NUMBER	Maximum number of user sessions logged on but inactive (not executing database commands)
SES_ACTIVE	NUMBER	Maximum number of user sessions logged on and active (executing database commands)
SES_WAITING	NUMBER	Maximum number of user sessions waiting for locks
SES_LOCKS	NUMBER	Maximum number of locks
SES_OPEN_CURSORS	NUMBER	Maximum number of open cursors
SES_DB_GETS	NUMBER	Total number of all db block gets for all user sessions since the database instance startup
SES_CPU	NUMBER	Total time in centiseconds (1/100 of seconds) spend by the database executing user commands since the database instance startup
SES_CPU_USED	NUMBER	Total time in centiseconds (1/100 of seconds) spend by the database executing user commands since the previous statistics collection job run.
SES_MEMORY	NUMBER	Total amount of SGA memory allocated to user sessions
WAIT_EVENT1_NAME	VARCHAR2(30)	Name of the first of top 5 wait events detected at the time of statistics collection job run.
WAIT_EVENT1	NUMBER	Average wait time in centiseconds (1/100 of seconds) for the first of top 5 wait events detected at the time of statistics collection job run.
WAIT_EVENT2_NAME	VARCHAR2(30)	Name of the second of top 5 wait events detected at the time of statistics collection job run.
WAIT_EVENT2	NUMBER	Average wait time in centiseconds (1/100 of seconds) for the second of top 5 wait events detected at the time of statistics collection job run.
WAIT_EVENT3_NAME	VARCHAR2(30)	Name of the third of top 5 wait events detected at the time of statistics collection job run.
WAIT_EVENT3	NUMBER	Average wait time in centiseconds (1/100 of seconds) for the third of top 5 wait events detected at the time of statistics collection job run.
WAIT_EVENT4_NAME	VARCHAR2(30)	Name of the fourth of top 5 wait events detected at the time of statistics collection job run.
WAIT_EVENT4	NUMBER	Average wait time in centiseconds (1/100 of seconds) for the fourth of top 5 wait events detected at the time of statistics collection job run.
WAIT_EVENT5_NAME	VARCHAR2(30)	Name of the fifth of top 5 wait events detected at the time of statistics collection job run.
WAIT_EVENT5	NUMBER	Average wait time in centiseconds (1/100 of seconds) for the fifth of top 5 wait events detected at the time of statistics collection job run.
WAIT_SEG1_NAME	VARCHAR2(171)	Name of the first of top 5 objects experiencing I/O contention (buffer busy waits) detected at the time of statistics collection job run. Name is reported in "owner.object (partition)" format for partitioned objects and in "owner.object" format for non-partitioned objects.
WAIT_SEG1	NUMBER	Maximum wait time in seconds for the first of top 5 objects experiencing I/O contention (buffer busy waits) detected at the time of statistics collection job run.
WAIT_SEG2_NAME	VARCHAR2(171)	Name of the second of top 5 objects experiencing I/O contention (buffer busy waits) detected at the


		time of statistics collection job run. Name is reported in "owner.object (partition)" format for partitioned objects and in "owner.object" format for no-partitioned objects.
WAIT_SEG2	NUMBER	Maximum wait time in seconds for the second of top 5 objects experiencing I/O contention (buffer busy waits) detected at the time of statistics collection job run.
WAIT_SEG3_NAME	VARCHAR2(171)	Name of the third of top 5 objects experiencing I/O contention (buffer busy waits) detected at the time of statistics collection job run. Name is reported in "owner.object (partition)" format for partitioned objects and in "owner.object" format for no-partitioned objects.
WAIT_SEG3	NUMBER	Maximum wait time in seconds for the third of top 5 objects experiencing I/O contention (buffer busy waits) detected at the time of statistics collection job run.
WAIT_SEG4_NAME	VARCHAR2(171)	Name of the fourth of top 5 objects experiencing I/O contention (buffer busy waits) detected at the time of statistics collection job run. Name is reported in "owner.object (partition)" format for partitioned objects and in "owner.object" format for no-partitioned objects.
WAIT_SEG4	NUMBER	Maximum wait time in seconds for the fourth of top 5 objects experiencing I/O contention (buffer busy waits) detected at the time of statistics collection job run.
WAIT_SEG5_NAME	VARCHAR2(171)	Name of the fifth of top 5 objects experiencing I/O contention (buffer busy waits) detected at the time of statistics collection job run. Name is reported in "owner.object (partition)" format for partitioned objects and in "owner.object" format for no-partitioned objects.
WAIT_SEG5	NUMBER	Maximum wait time in seconds for the fifth of top 5 objects experiencing I/O contention (buffer busy waits) detected at the time of statistics collection job run.
IO_APPL1	VARCHAR2(30)	Name of the first of top 5 applications causing lots of db buffer gets detected at the time of statistics collection job run.
IO_APPL1_COUNT	NUMBER	Number of running instances of the first of top 5 applications causing lots of db buffer gets detected at the time of statistics collection job run.
IO_DB_GETS1	NUMBER	Total number of all db buffer gets for the first of top 5 applications causing lots of db buffer gets (all running instances).
IO_APPL2	VARCHAR2(30)	Name of the second of top 5 applications causing lots of db buffer gets detected at the time of statistics collection job run.
IO_APPL2_COUNT	NUMBER	Number of running instances of the second of top 5 applications causing lots of db buffer gets detected at the time of statistics collection job run.
IO_DB_GETS2	NUMBER	Total number of all db buffer gets for the second of top 5 applications causing lots of db buffer gets (all running instances)
IO_APPL3	VARCHAR2(30)	Name of the third of top 5 applications causing lots of db buffer gets detected at the time of statistics collection job run.
IO_APPL3_COUNT	NUMBER	Number of running instances of the third of top 5 applications causing lots of db buffer gets detected

		at the time of statistics collection job run.
IO_DB_GETS3	NUMBER	Total number of all db buffer gets for the third of top 5 applications causing lots of db buffer gets (all running instances).
IO_APPL4	VARCHAR2(30)	Name of the fourth of top 5 applications causing lots of db buffer gets detected at the time of statistics collection job run.
IO_APPL4_COUNT	NUMBER	Number of running instances of the fourth of top 5 applications causing lots of db buffer gets detected at the time of statistics collection job run.
IO_DB_GETS4	NUMBER	Total number of all db buffer gets for the fourth of top 5 applications causing lots of db buffer gets (all running instances).
IO_APPL5	VARCHAR2(30)	Name of the fifth of top 5 applications causing lots of db buffer gets detected at the time of statistics collection job run.
IO_APPL5_COUNT	NUMBER	Number of running instances of the fifth of top 5 applications causing lots of db buffer gets detected at the time of statistics collection job run.
IO_DB_GETS5	NUMBER	Total number of all db buffer gets for the fifth of top 5 applications causing lots of db buffer gets (all running instances).
CPU_APPL1	VARCHAR2(30)	Name of the first of top 5 applications taking most CPU resources detected at the time of statistics collection job run.
CPU_APPL1_COUNT	NUMBER	Number of running instances of the first of top 5 applications taking most CPU resources detected at the time of statistics collection job run.
CPU_USED1	NUMBER	Total number of all db buffer gets for the first of top 5 applications taking most CPU resources (all running instances).
CPU_APPL2	VARCHAR2(30)	Name of the second of top 5 applications taking most CPU resources detected at the time of statistics collection job run.
CPU_APPL2_COUNT	NUMBER	Number of running instances of the second of top 5 applications taking most CPU resources detected at the time of statistics collection job run.
CPU_USED2	NUMBER	Total number of all db buffer gets for the second of top 5 applications taking most CPU resources (all running instances).
CPU_APPL3	VARCHAR2(30)	Name of the third of top 5 applications taking most CPU resources detected at the time of statistics collection job run.
CPU_APPL3_COUNT	NUMBER	Number of running instances of the third of top 5 applications taking most CPU resources detected at the time of statistics collection job run.
CPU_USED3	NUMBER	Total number of all db buffer gets for the third of top 5 applications taking most CPU resources (all running instances).
CPU_APPL4	VARCHAR2(30)	Name of the fourth of top 5 applications taking most CPU resources detected at the time of statistics collection job run.
CPU_APPL4_COUNT	NUMBER	Number of running instances of the fourth of top 5 applications taking most CPU resources detected at the time of statistics collection job run.
CPU_USED4	NUMBER	Total number of all db buffer gets for the fourth of top 5 applications taking most CPU resources (all running instances).
CPU_APPL5	VARCHAR2(30)	Name of the fifth of top 5 applications taking most CPU resources detected at the time of statistics collection job run.

CPU_APPL5_COUNT	NUMBER	Number of running instances of the fifth of top 5 applications taking most CPU resources detected at the time of statistics collection job run.
CPU_USED5	NUMBER	Total number of all db buffer gets for the fifth of top 5 applications taking most CPU resources (all running instances).
LAST_SCN	NUMBER	Last system change number (SCN) recorded before statistics collection job run. Note: Oracle increments SCN every time a transaction is committed.
LAST_SCN_ARCHIVED	NUMBER	Last archived system change number (SCN) before statistics collection job run. Note: Oracle increments SCN every time a transaction is committed. The archived SCN represents number of last transaction archived in archive log files.
SCH_ARCHIVE_RATE	NUMBER	System archival rate (transactions/hour) - number of archived transactions between 2 statistics collection job runs.

Monitor connections and resources (Monitor menu)

You can use monitors to gather lots of database characteristics in a real-time. All monitors query dynamic performance tables in combination with data dictionary tables and views.

To customize monitors behavior such as [sampling cycle](#) (refresh rate), select **Tools/Options** command or simply click the Options button on the [toolbar](#) . An [Option dialog](#) box will appear. For your convenience, active monitor properties will be preloaded so you can quickly change them. DB Monitor Expert allows you easily change [sorting](#) and [filtering](#) by clicking on monitor column headers and selecting commands from **View** menu. You can also right-click any monitor to get some additional features accessible from a popup menu that appears in response to the click.

DB Monitor Expert provides the following real-time monitors:

Connections

Use this monitor to track individual user processes and process parameters. This monitor displays cumulative information about user connections including connection (session) status, the application and terminal from which the connection was established.

To drill-down user activity:

- Select **File/Memory and CPU Usage** to track current and top session memory.
- Select **File/Query** text to see last SQL statement executed by the session.
- Select required session on the **Connections** tab then click on the **Session Details** tab to see in depth all session parameters.
- Select the **SQL** tab to see session's SQL statements. SQL tab on refresh automatically calculates timing, affected data and other statistics for the displayed SQL statements. Keep this tab page open to allow persistent SQL tracking. New SQL statements appear at the end of the view. Note that the monitor is discreet in time. It cannot catch SQL statements that were executed during sampling cycle. Specify fast refresh interval to improve accuracy of the SQL view.

Open Cursors

A cursor is a handle for the memory associated with a specific statement. ORACLE logs all statements that are being executed. Looking at open cursors display you can find out what your database is doing now and find out user that initiated executed statements. Open cursors monitor also helps to trace system-wide SQL activity. Note that only first 60 characters of each SQL statement are displayed. To find out full text of executed statements use the SQL Area monitor.


SQL Area

SQL Area is a part of the shared pool that holds shared SQL and PL/SQL statements (referred as *the library cache*), dictionary cache, and information on sessions against the database. You control the size of the shared pool by setting SHARED_POOL_SIZE parameter in INIT.ORA. SQL statements are placed in the shared pool and flushed according to an LRU (least recently used) algorithm.

By monitoring SQL Area, you can see what statements are sitting there. Knowing what statements are already there assists you in coding SQL that matches.

Look at reloads column to find out number of misses. If the ration of reloads to executions is greater than 1%, consider adding memory to the shared SQL pool. Also, look at the offensive statements that have high percent of disk reads and/or buffer gets. They appear in red.

From SQL Area monitor you can invoke an [Explain Plan](#) for any SELECT statement. Do the following:

- 1 Highlight the statement for which you want to get plan explanation (click on the first line of the sell containing SQL text or on any other sell in the same row). The highlighted sell has black background.
- 2 Select **File/Explain Plan** command or simply click the Explain Plan button on the [toolbar](#) .
- 3 DB Monitor Expert will retrieve enhanced **Explain Plan**.

Object Access

The object access monitor shows tables, procedures, views, etc that are in use and the users and programs accessing them. You can use filtering and sorting facilities for limiting display to specific session, user or program.

Cached Objects

Use cached objects display for monitoring library cache. You can use filtering and sorting facilities for easy finding most popular objects that were loaded/executed. For instance, you can do descendant sort by loads and filter out objects that have `type='TABLE'`. This will tell you most frequently accessed tables.

Application Effectiveness

By looking at the table scans (short tables) and table scans (long tables) rows, you can determine how many full table scans your database is doing. A *full table scan* occurs when ORACLE does not have an appropriate index to use, and it is forced to read every row in a table. If you intend to process the majority of records, then using a full table scan is faster. In most environments, though, expect to see the majority of the access being done via indexes. So, look at non-index lookups ratio to determine if there is a problem.

File Input/Output

Use file I/O to observe both cumulative physical reads and writes activity by file. By analyzing this information, you can pinpoint which files are used most heavily, making sure that they are on separate disks and separate disk controllers. Detect files that have an uneven work load and try to balance a distribution of I/O by doing following:

- Move one or more database files to another disk to balance the load.
- When a frequently accessed table shares space in a data-file with other tables, take the busy table out of the shared data-file and put it in its own data-file on another drive.

Rollback Segments

Examine rollback segments display to determine if there is contention for rollback segments. If rollback segment waits/gets percent for any segment (number of header waits divided by number of header gets) is

greater than 10%, additional rollback segments may be required, especially if the number of active transactions is significant for this rollback segment. To examine rollback segment activity you can use Rollback Transaction monitor (see next section). To examine rollback waits statistic use the Waits monitor.

Rollback Transactions

This monitor simply displays active rollback segment transaction and processes that own those transactions.

Buffer Cache

Database buffers of the SGA store the most recently used blocks of database data. Remember that all data must pass through this cache before they can be accessed by ORACLE. Use buffer cache monitor to track cumulative physical and logical reads from buffer cache. Look at hit ratio that measures effectiveness of the cache. Usually a healthy database has hit ratio between 90% and 95%. You can set `DB_BLOCK_BUFFERS` parameter in `INIT.ORA` to adjust size of the buffer cache.

Dictionary Cache

Dictionary cache monitor tracks total misses vs. gets ratio for database dictionary cache. Usually that ratio falls below 7%. If it is greater than 7% consider increase of `SHARED_POOL_SIZE` parameter in `INIT.ORA`. A poorly tuned dictionary cache can degrade performance by 50% or even more.

Redo Log Latches

Latches are low-level locks on Oracle shared internal structures. The redo log latches monitor displays redo log latch activity statistics for processes of the database instance.

Recursive Calls

Recursive calls are reads of data dictionary information from disk that is not in the SGA. Recursive calls ratio is calculated as average number of recursive calls per number of opened cursors. Recursive calls ratio is usually between 7.0 and 10.0 for tuned production systems and might be as high as 10.0 - 14.5 for tuned development systems. Also, look at parse count ratio calculated as average number of parses per number of opened cursors. Parse count ratio usually falls between 1.15 and 1.45. If it is higher then it is usually a sign of not optimized SQL queries.

Dispatcher Processes

The Dispatcher monitor displays summary information about dispatcher processes. All processes are grouped by network protocol.

Sorts

The *sorts (disk)* row tells you how many times you had to sort to disk; that is size that could not be handled by the size of `SORT_AREA_SIZE` parameter in `INIT.ORA`. The *sorts (memory)* row tells you how many times you were able to complete the sort using just memory. Usually 90% or higher of all sorts should be done in memory.

Waits

Use waits monitor to find out all waits for resources. A majority of waits usually related with rollback segments activity. If your waiting is high, try to add more rollback segments. On other hand, if your waiting was due to data blocks, you need to modify your *freelist* settings for most busy tables.

Locks

Oracle establishes different forms of locking to insure consistency, concurrency and integrity of its data. In addition, locks can be issued "manually" during database application operations. Use locks monitor to detect abnormal or unnecessary locking situations and requests. Sort locks by objects for easy finding dead locks and resolving sessions blocked by locks.



SGA

Use this display to get picture of SGA (System global Area) memory distribution. Compare total memory allocated by SGA to the amount of memory available on the server. Keep in mind, that SGA should never take over 50% of the available memory, otherwise the low memory impact will slow down the entire system.

Custom Monitors

In addition to the variety of predefined monitors and customizing features that DB Monitor Expert provides, you can also define your own monitors. User-defined monitors can address your database specific requirements. Using custom monitors, you can also automate problem detection and correction, and optionally execute a job to fix the problem and generate alarm message that may be sent via E-mail, pager or beeper to you or other people.

To create a new monitor:

1. Select **Monitor/Custom Monitor** menu. DB Monitor Expert will open the Custom Monitors dialog. The dialog window will display all already created custom monitors.
2. Click the **New** button.
3. Enter valid name in the **Name** field on the **Properties** tab. Note that name may consist spaces.
4. Specify appropriate timer interval in the **Interval** field. The number must be in range from 1 to 32767 seconds.
5. Type valid SQL SELECT statement in the **Select** field. Alternatively, you can paste SQL from the clipboard. To do this, use keyboard shortcut CTRL V or use right mouse click on the **Select** field then from popup menu choose the **Paste** command. Also, you can click on the **Open File**  button to paste SQL from the file. This SQL statement will be used to gather statistics to be displayed in the monitor you are defining. Make sure that the SQL statement does not end with a semicolon (;). To avoid unclear column names in the monitor syntax always use aliases in the select statement (see [examples](#)).
6. Click on the **Preview** tab to customize display properties of the monitor. You can resize columns, set filter and additional sort order. See [filtering](#) and [sorting](#) topics for details. Note that if later change the SELECT statement you will need to re-apply design changes again.
7. Click the **Filter** button to specify additional Filter that DB Monitor Expert will apply after retrieving data for the monitor.
8. Click the **Sort** button to change data order after DB Monitor Expert retrieve data for the monitor.
9. Click on the **Threshold** tab to customize new monitor behavior. If required, enter threshold condition that the monitor will verify every time a new sampling will occur.
10. If you wish, you can specify *fixit* job, which the monitor performs when the threshold condition is detected. In the **SQL File** field type SQL file name where *fixit* job specified. Alternatively, you can click **Open File**  button to browse and paste the SQL file name. For example, you can have the monitor for monitoring the tablespace TEMP in the database SALES and generate an alert if the tablespace runs out of free space. You can also have a data-file automatically added to the tablespace if this event occurs.
11. To generate alert message, fill all fields in the **Action** box. Following methods are presently supported: [E-mail](#), [Pager](#), and [Beeper](#). Note that the Beeper method is provided so that you can send messages to "Numeric Only" pagers. It does not use the TAP protocol. DB Monitor Expert simply takes the phone line off-hook, dials the pager phone number, waits x number of seconds, touch tones the digits to be displayed on the pager and hangs up. It is recommended that

- whenever possible you use other described methods to send alert messages, because of their reliability.
12. Click the **Save** button to save the new monitor definition. The monitor can be used again later on.
 13. Click the **Run** button to start the monitor you have defined. Note that you can run the monitor without doing save first, but by default DB Monitor Expert does not save the monitor definition for you until you click the **Save** button. So that if you do not save the definition you will not be able to reuse this monitor later.
 14. Now you can build another monitor or click the **Exit** button to close Custom Monitor dialog.

The following example of user-defined monitor is included in the standard installation package:

Free space monitor

```

SELECT seg.owner,
       seg.segment_name,
       seg.segment_type,
       seg.tablespace_name,
       to_char(seg.max_extents, '999,999,999') next_extent
FROM sys.dba_segments seg
WHERE seg.extents = seg.max_extents

UNION ALL

SELECT seg.owner,
       seg.segment_name,
       seg.segment_type,
       seg.tablespace_name,
       to_char(t.next_extent, '999,999,999')
FROM sys.dba_segments seg,
     sys.dba_tables t
WHERE seg.segment_type = 'TABLE'
      AND seg.segment_name = t.table_name
      AND seg.owner = t.owner
      AND NOT EXISTS (SELECT tablespace_name
                     FROM dba_free_space free
                     WHERE free.tablespace_name = t.tablespace_name
                           AND free.bytes >= t.next_extent
                     )

UNION ALL

SELECT seg.owner,
       seg.segment_name,
       seg.segment_type,
       seg.tablespace_name,
       to_char(i.next_extent, '999,999,999')
FROM sys.dba_segments seg,
     sys.dba_indexes i
WHERE seg.segment_type = 'INDEX'
      AND seg.segment_name = i.index_name
      AND seg.owner = i.owner
      AND NOT EXISTS (SELECT tablespace_name
                     FROM dba_free_space free
                     WHERE free.tablespace_name = i.tablespace_name
                           AND free.bytes >= i.next_extent
                     )

UNION ALL

SELECT seg.owner,
       seg.segment_name,
       seg.segment_type,
       seg.tablespace_name,
       to_char(c.next_extent, '999,999,999')
FROM sys.dba_segments seg,
     sys.dba_clusters c
WHERE seg.segment_type = 'CLUSTER'
      AND seg.segment_name = c.cluster_name
      AND seg.owner = c.owner

```

```

        AND NOT EXISTS (SELECT tablespace_name
                        FROM dba_free_space free
                        WHERE free.tablespace_name = c.tablespace_name
                           AND free.bytes >= c.next_extent
                        )
    UNION ALL

    SELECT seg.owner,
           seg.segment_name,
           seg.segment_type,
           seg.tablespace_name,
           to_char(r.next_extent, '999,999,999')
    FROM sys.dba_segments seg,
         sys.dba_rollback_segs r
    WHERE seg.segment_type = 'ROLLBACK'
           AND seg.segment_name = r.segment_name
           AND seg.owner = r.owner
           AND NOT EXISTS (SELECT tablespace_name
                           FROM dba_free_space free
                           WHERE free.tablespace_name = r.tablespace_name
                              AND free.bytes >= r.next_extent
                           )

```

See also:

[Historical trends - database performance](#)
[Graph - dynamic statistic](#)
[Reports](#)
[Data manipulation and data export](#)
[Customization](#)

Using Thresholds in Custom Performance Monitors

Normally threshold values in custom graphical performance monitors should be used with one or more communications options. Using these options you can configure DB Monitor Expert to notify you whenever the specified threshold value is reached.



Important Notes: In order to detect a threshold-reached event value DB Monitor Expert must be running at that time and the graphical monitor must open on the screen. Although graphical custom monitors with threshold values are very easy to build they are applicable as a solution to full time performance monitoring! Use them as a quick way to monitor things while developing new applications or making database changes. As a long-term solution for building always-on performance monitors use custom procedures running on the back-end and using [DB Alert Expert](#) to communicate to you using email, pager or other communication methods.

The following 3 communication methods are supported in custom performance monitors:

1. [Sending Messages To Alpha-numeric Pagers](#)
2. [Sending Email Messages](#)
3. [Sending Messages to Alpha-numeric and Numeric Beepers](#)

Sending Messages to Pagers




Note: Before attempting to send a direct page, please check with the intended recipient to make sure they have registered their pager with CompuServe or other similar service.

There is no principal difference in sending [E-mail](#) messages and sending messages to a pager. The only difference is that when you send a message to a pager you must use a special addressing format. For example, the address format for the CompuServe is:


[CompuServe_ID@mobile.compuserve.com](#)

For example, if the account number (CompuServe ID) is 79999,9999, recipient could receive wireless e-mail by using the address "79999.9999@mobile.compuserve.com". Note that period must be used in place of comma.

 **Tip:** In the case of an alphanumeric message, the number of characters received by the recipient depends on how they set up their pager preferences. Keeping your message short and concise helps to ensure that the recipient gets the most out of your message.

See [E-mail](#) topic for detail on sending email messages.

Pager Services FAQ

 **Note:** You need CompuServe or similar account in order to use paging services.

Q1. Who are the Partner Carriers?

A1. The Partner Carriers are:

- . Air Media
- . MobileComm
- . PageMart
- . PageNet
- . SkyTel

Q2. How do I register my pager with CompuServe?

A2. To register your pager, you can select GO PAGER and select "Register My Pager". There is a one-time fee of \$5.00 to register your pager with CompuServe.

Q3. How do I validate my pager?

A3. When you receive a validation message on your pager, it will include a validation number. Once you have the number, follow these steps to complete registration:

1. Select GO PAGER
2. Click on "Register My Pager"
3. Click on "Devices/Preferences"
4. Click on "Validate Device"

Q4. Is there a pager registration support number?

A4. YES, the support number, 1-800-560-0750, can be used to get initial Pager registration support. They are very helpful with validation and pager modem numbers.

Q5. How do I change my pager options/preferences?

A5. GO PAGER and select "Change Current Pager Preferences", "Device/Preferences", "Review/Change Preference". You may set the number of pages to receive per day/month, the number of characters per message, subject, or originator area.

Q6. How does someone address messages from within CompuServe to be sent to my pager?

A6. Messages originating from another CompuServe member simply need a "mobile" prefix to your existing CompuServe address. For example, if your CompuServe address is "79999,999" your pager address will be "mobile:79999.999" (**Note:** you must change the comma to a period). Personal Addresses also take the same "mobile:" prefix (i.e. mobile:johndoe).

Q7. How does someone on the Internet address messages to go to my pager?

A7. Messages originating from the Internet need to include "mobile" as a prefix to the CompuServe domain name. For example, if your CompuServe address is "79999.999" your pager address will be "79999.999@mobile.compuserve.com" (**Note:** you must change the comma to a period). Personal Addresses also take the same "mobile" prefix -- i.e. johndoe@mobile.compuserve.com

Q8. Why did my messages/scores/quotes suddenly stop going to my pager?

A8. You may have reached your daily or monthly limit of messages set in your pager options. To check your status, select GO PAGER, "Change Current Pager Preferences", "Devices/Preferences" and then "Device Status".

Q9. How do I disconnect my pager from CompuServe?

A9. Enter the GO PAGER command. Click "Register my Pager on CompuServe." Click on "Devices/Preferences." Click on "Remove Device." NOTICE: If you select Remove Device and then register your pager again at a later time, you will be charged the \$5.00 registration fee for each time you select to register your pager.

Q10. What safeguards are in place?

A10. Several security measures are built into the pager registration process. These safeguards protect against sending and receiving messages to the wrong pager and supplement the standard security provided by your account password:

- * First, when you register a pager, the required information includes the pager number (which is printed on the back of the pager itself). This number is hard-coded into the circuitry of the pager, and it serves as a PIN number similar to one you would use in an ATM transaction.

- * Next you receive a validation message with a randomly-generated number.

- * Then you use the GO PAGER command to enter the number from the validation message.

Sending E-Mail Messages

DB Monitor Expert supports only Messaging Application Program Interface (MAPI). Before attempting to send a message, please make sure you e-mail system is MAPI compliant.


You need to create a [user-defined monitor](#) before you can specify an event that must be detected by that monitor and alert message is generated. Once you created such monitor, you should fill out the following fields on the **Threshold** tab in order to send e-mail message:

Field	Meaning
Method	Communication method. Select E-mail from drop-down list.
User ID	A string whose value is the mail system user profile name. For example "Exchange Settings".
Password	A string whose value is the mail system password.
Recipient	A string whose value is the recipient's mail address. Note that your mail system may support comma-separated list of recipients.
Message	A string whose value is the actual message.

Sending Messages to Beepers

You must have a modem in order to send numeric messages to a beeper. You need to create a [user-defined monitor](#) before you can specify an event that must be detected by that monitor and alert message is generated. Once you created such monitor, you should fill out the following fields on the **Threshold** tab in order to send numeric message to a beeper:

Field	Meaning
Method	Communication method. Select Beeper from drop-down list.
Number	A number whose value is the recipient's beeper number. Note that you can use spaces, parenthesis, and commas in the number. A comma causes modem to pause when dialing the number.
Message	A number whose value is the actual message. Note that alphanumeric messages are not supported in this version of the DB Monitor Expert.

 **Warning:** You have to specify your modem properties on the [Modem page](#) in the system [Options](#) before you can send a message.

Performance Library

About the Performance Library


The Performance Library is a collection of over 50 pre-built ready-to-use monitors, graphs, and reports that can be found in the DB Monitor Expert library of user-defined objects.

To access Performance Library objects use the following commands from the DB Monitor Expert menu:

- Monitor/Custom Monitor
- Graph/Custom Graph
- Trends/Custom Trend Graph
- Reports/Custom Reports

Using DB Monitor Expert menus you can also add to the Performance Library your own monitors, graphs, and reports. For details on how to create custom objects in the Performance Library see [Custom Monitor](#), [Custom Graph](#), [Custom Trend Graph](#), and [Custom Report](#) topics.

Enhanced Explain Plan

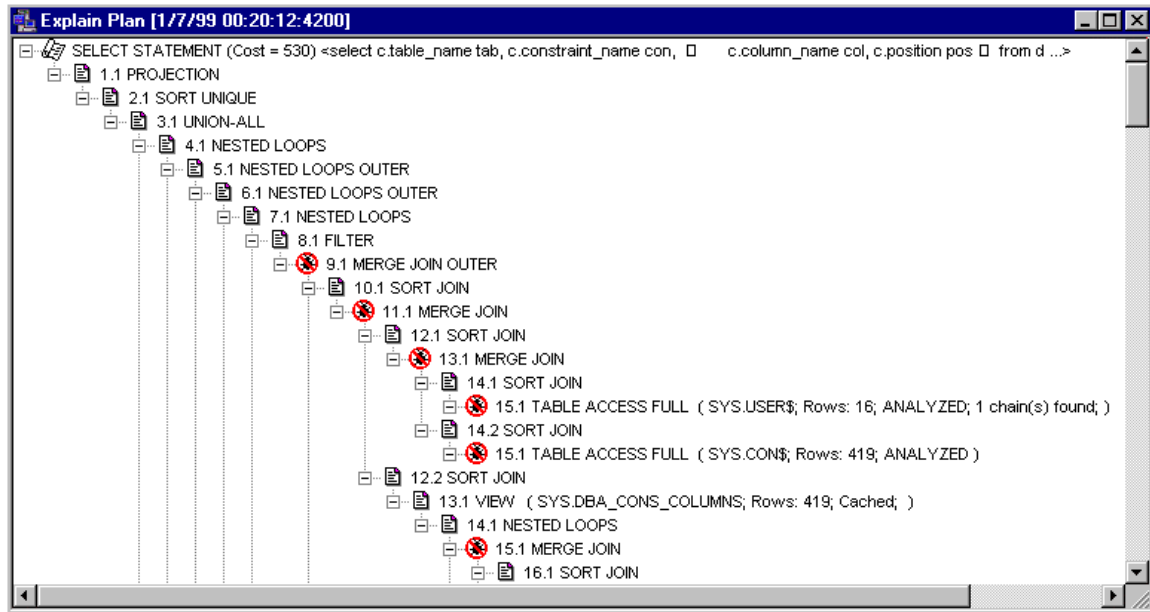
Both SQL Editor and DB Monitor Expert include visual **Explain Plan** utility, which provides SQL execution plan in a user-friendly graphical format. Explain Plan is an important tool for analyzing the efficiency of access paths used in your queries. To get execution plan in the [DBA Notepad](#) window type SQL statement for which you want to get explanation of the execution strategy and select **File/Explain Plan** command or simply click the Plan button on the [toolbar](#). 

The output shows how DBMS will execute the analyzed SQL statement, including:

- What database operations are being performed in order to execute the statement
- Which, if any, indexes are being used
- Whether or not the proper set of columns are included in your indexes
- How many rows exist in the affected tables
- Whether or not the affected tables and clusters have chained rows
- Whether or not the affected objects have been analyzed so that the object statistics are available to the Oracle Query Optimizer.

For detailed information on the specific meaning Explain Plan output, consult the Oracle documentation.

In order to archive better plan you can use various tips and techniques fully described in the Oracle documentation.



Object analysis tools

DB Monitor Expert provides one-step method for object analysis. To perform batch analyses select one or more objects (tables/indexes/cluster) you want to analyze and choose **File/Analyze** command). Regularly performed analyzes help the database optimizer to choose right [execution plan](#) for the SQL statements to the database from all database application. Use tables/indexes/cluster analyze to improve performance of your application especially after significant changes were made in the table data or structure.

In order to analyze a table, index or cluster, it must belong to your schema or you must have **ANALYZE ANY** privileges.



Tip: Pressing hot key CTRL + will select all available objects for analysis. Pressing CTRL - will deselect all previously selected objects.

See also:

- [Table Analyzer](#)
- [Index Analyzer](#)
- [Cluster Analyzer](#)

Table Analyzer

DB Monitor Expert allows you to analyze one or more tables to compute or estimate statistics stored in the Oracle data dictionary. In addition you can validate the structure of the table(s) and its indexes and identify chained rows of the table(s).

In order to analyze a table, it must belong to your schema or you must have **ANALYZE ANY** privileges.

Analysis Type	Explanation
Compute Statistics	Computes exact statistics about the table for use by the optimizer then stores them in the system catalog.

Estimate Statistics For First <percent> of rows	Estimates statistics about the table for use by the optimizer and stores them in the system catalog. The statistics is based on specified percent of rows as a sample. Estimate statistics method is a faster but less thorough alternative to computing statistics.
Estimate Statistics For First <number> of rows	Estimates statistics about the table for use by the optimizer and stores them in the system catalog. The statistics is based on specified number of rows as a sample. Estimate statistics method is a faster but less thorough alternative to computing statistics.
Validate Structure	Validates the structure of the table. If you choose the Cascade option, then it also validates the table's indexes.
Cascade Indexes	Validates the structure of the table and the table's indexes.
Analyze and Fix Chained Rows	Identifies any migrated or chained rows of data in the table. The result information is placed into CHAINED_ROWS table. If the table does not exist then DB Tools asks your permissions to create one. In case if chained rows are found, DB Monitor Expert can fix them by moving chained rows into a temporary table then deleting and reinserting these rows into the table being analyzed.

To analyze a table:

1. Select **Tools/Table Analyzer** command. [Select Schema dialog](#) will appear.
2. Select schemas for which you want tables to be analyzed.
3. Highlight one or more tables to be analyzed and select **File/Analyze** command (shortcut: CTRL O) The Analyze dialog will open.
4. Specify the analysis type. Click the OK button.



Note: For large tables, **Analyze** operation may take very long time. To reduce that time choose **Estimate Statistics** For First <number> of rows.

See also:

[Index Analyzer](#)
[Cluster Analyzer](#)

Index Analyzer

DB Monitor Expert allows you to analyze one or more table indexes to compute or estimate statistics stored in the Oracle data dictionary. In addition, you can validate the structure of indexes.

In order to analyze an index, it must belong to your schema or you must have **ANALYZE ANY** privileges.

Analysis Type	Explanation
Compute Statistics	Computes exact statistics about the index for use by the optimizer then stores it in the system catalog.
Estimate Statistics For First <percent> of rows	Estimates statistics about the index for use by the optimizer and stores them in the system catalog. The statistics is based on specified percent of rows as a sample. Estimate statistics method is a faster but less thorough alternative to computing statistics.
Estimate Statistics For First <number> of rows	Estimates statistics about the index for use by the optimizer and stores them in the system catalog. The statistics is based on specified number of rows as a sample. Estimate statistics method is a faster but less thorough alternative to computing statistics.
Validate Structure	Validates the structure of the index.

To analyze an index

1. Select **Tools/Index Analyzer** command. [Select Schema dialog](#) will appear.
2. Select schemas for which you want indexes to be analyzed.

3. Highlight one or more indexes to be analyzed and select **File/Analyze** command (shortcut: CTRL O) The Analyze dialog will opens.
4. Specify the analysis type. Click the OK button.



Note: For large tables, Analyze operation may take very long time. To reduce that time choose **Estimate Statistics For First <number>** of rows.

See also:

[Table Analyzer](#)
[Cluster Analyzer](#)

Cluster Analyzer

DB Monitor Expert allows you to analyze one or more clusters to compute or estimate statistics stored in the Oracle data dictionary. In addition you can validate the structure of the cluster(s) and its indexes and identify chained rows of the cluster table(s).

In order to analyze a cluster, it must belong to your schema or you must have **ANALYZE ANY** privileges.

Analysis Type	Explanation
Compute Statistics	Computes exact statistics about the cluster for use by the optimizer then stores it in the system catalog.
Estimate Statistics For First <percent> of rows	Estimates statistics about the cluster for use by the optimizer and stores them in the system catalog. The statistics is based on specified percent of rows as a sample. Estimate statistics method is a faster but less thorough alternative to computing statistics.
Estimate Statistics For First <number> of rows	Estimates statistics about the cluster for use by the optimizer and stores them in the system catalog. The statistics is based on specified number of rows as a sample. Estimate statistics method is a faster but less thorough alternative to computing statistics.
Validate Structure	Validates the structure of the cluster. If you choose the Cascade option, then Oracle will also validates indexes of cluster.
Cascade Indexes	Validates the structure of the table and the table indexes.
Analyze and Fix Chained Rows	Identifies any migrated or chained rows of data in the cluster. The result information is placed into CHAINED_ROWS table. If the table does not exist then DB Monitor Expert asks your permissions to create one. In case if chained rows are found, DB Monitor Expert can fix them by moving chained rows into a temporary table then deleting and reinserting these rows into the table being analyzed.

To analyze a cluster

1. Select **Tools/Cluster Analyzer** command. [Select Schema dialog](#) will appear.
2. Select schemas for which you want clusters to be analyzed.
3. Highlight one or more clusters to be analyzed and select **File/Analyze** command (shortcut: CTRL O) The Analyze dialog opens.
4. Specify the analysis type. Click the OK button.




Note: For large tables, **Analyze** operation may take very long time. To reduce that time choose **Estimate Statistics For First <number>** of rows.

See also:

[Table Analyzer](#)
[Index Analyzer](#)


Scheduler

To open the Scheduler graphical interface select **Tools/Scheduler** command, or alternatively click Scheduler button  on the [application toolbar](#).

Do not confuse the scheduler available in DB monitor Expert with the DB Job Expert component!

The Scheduler supports both client side and server side jobs. For details on server side jobs see description of the DBMS_JOB package in the Oracle Administration Guide. To use server side scheduling and job execution you must have sufficient authority to access SYS.DBMS_JOB package. Contact your Oracle DBA if you do not have enough authority.


Scheduling a new job:

- 1 Run the Scheduler by clicking on the Scheduler icon (see instructions above)
- 2 You can schedule simultaneously one SQL script, one report, and one external program.
 - **To schedule SQL script:** Type the date and time when you want to run the job in the **Start** field of the **SQL script** section. You may use scroll bars to adjust start date and time. Note that adjustment is performed to the current section (where cursor is) of the date-time field. Scrolling insures that you do not enter invalid date or time such as February 30. Press Tab to move to the next field. Type in the name of the SQL file in the SQL script edit field. Use the **Browse** button  to access the standard Browse facility to search for and select a file. Note that file existence is not verified when you schedule the job. DB Monitor Expert checks file when time to run the job comes. This flexibility allows you to produce/modify file from another program from time to time.

The SQL script may have one or more SQL statements in. If the script features SELECT statements and you wish to print their result sets, check **Print results** box.


To repeatedly run scheduled SQL script, scroll the **Repeat every** field until you find appropriate time interval. If you do not wish to repeat the SQL script, make sure you have **[None]** in **Repeat script every** field.

- **To schedule report:** Type the date and time when you want to run the job in the **Start** field of the **Report** section. You may use scroll bars to adjust start date and time. Note that adjustment is performed to the current section (where cursor is) of the date-time field. Scrolling insures that you do not enter invalid date or time such as February 30. Press Tab to move to the next field. Select from drop down list report name you wish to run. Note that report names are transparent to name of commands under application **Monitor** and **Graph** menus. You also can schedule custom monitors, graphs, and reports you created using **Monitor/Custom Monitor**, **Graph/Custom Graph**, and **Report/Custom Report** commands. For more details see [Custom Monitor](#), [Custom Graph](#), and [Custom Report](#) topics. You must use unique names for your custom objects, otherwise the Scheduler finds first matching name and submits the job.

Click properties button  to specify custom filter and sort criteria for the scheduled report. You are not allowed to specify another filter or sort for user-defined reports, monitors and graphs, modify your custom object definition instead.

To repeatedly run scheduled report, scroll **Repeat every** field until you find appropriate time interval. If you do not wish to repeat the report, make sure you have **[None]** in **Repeat report every** field.

- **To schedule external program:** Type the date and time when you want to run the job in the **Start** field of the **External program** section. You may use scroll bars to adjust start date and time. Note that adjustment is performed to the current section (where cursor is) of the date-time field. Scrolling

insures that you do not enter invalid date or time such as February 30. Press Tab to move to the next field. Type in the name of the program file in the Program Name edit field. Use the **Browse** button  to access the standard Browse facility to search for and select a program file. Note that file existence is not verified when you schedule the job. DB Monitor Expert checks file when time to run the job comes. This flexibility allows you to produce/modify file from another program from time to time.

To repeatedly run scheduled program, scroll **Repeat every** field until you find appropriate time interval. If you do not wish to repeat the program, make sure you have **[None]** in the **Repeat every** field.

Connecting scheduler to the database.

If you schedule a long waiting task and do not want to keep connection to the database until time to run scheduled task, check **Auto-connect/disconnect** box. You must be connected to the database at the time when you set scheduler for auto-connect options, because the scheduler picks up from connection context all necessary connection properties. When it is time to run SQL script or report the schedule will establish connection to the database, run scheduled job, then disconnect from the database.

Locking DB Monitor Expert console when you are not around.

If you schedule a long waiting task and you do not want anybody to use DB Monitor Expert, you can check **Lock console** box to lock the program after you finish scheduling. To unlock the program, hit any key. **Enter user name and password** dialog will appear. Enter the same name and password you use to establish connection to the database. Click the **OK** button.

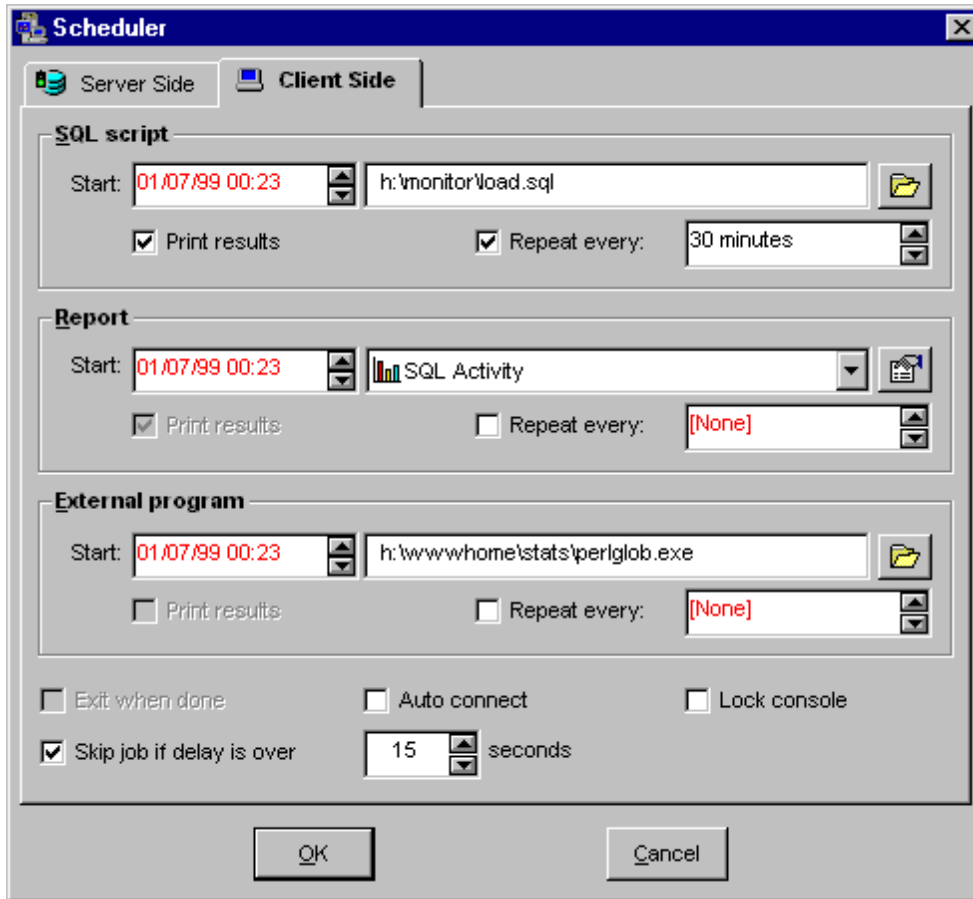
You can have DB Monitor Expert quit automatically after all scheduled jobs are done.

You should not schedule repeatable jobs in this case; otherwise, it will never stop.



Notes:

- You can schedule simultaneously any combination of SQL script, report and external program. However, when their execution times are equal DB Monitor Expert executes scheduled SQL script first, then scheduled report and last - external program.
- If "skip expired jobs" option is specified and execution of jobs that have higher priority takes longer than specified allowed delay, DB Monitor Expert does not execute expired jobs.
- The scheduler displays on the [status bar](#) the countdown time for the next scheduled job.
- If any of scheduled jobs fail, all other pending jobs will be suspended.



To Stop a Job:

- 1 Run the Scheduler by clicking on the Scheduler icon (see above)
- 2 Either enter zero start time (00/00/00 00:00) or remove job from job field.
 - **To remove SQL script:** empty **SQL script file** field.
 - **To remove report:** select blank name from the **Reports** drop down list box.
 - **To remove external program:** empty program file field.
- 3 When you are finished, click the OK button.

Connection Manager

DB Monitor Expert includes a basic process monitor to assist you in managing server activity. To access the Connection Manager select **Tools/Connection Manager** (shortcut: ALT T, M) from the application menu.


The Connection Manager's window consists of the following three tab pages:

Connections - Displays all database sessions. All active sessions have session status **ACTIVE** in red. All inactive sessions have **INACTIVE** status in black. Choose **Show inactive only** option on top of the Connection Manager window to filter inactive sessions only. Choose **Show active only** option on top of the Connection Manager window to filter active sessions only.

Waiting Processes - Displays all database sessions that are locked by other sessions and waiting for locks to be released.

Lock Tree - Displays for all database locks, including pending locks in the tree-view style structure. Hold locks appear on a first level and have yellow hand icon. Locks that are pending and waiting for other locks to be released first appear as second level leafs (white hand icon). Detected deadlocks have red hand icon and bold text.

To change refresh rate:

1. Select **Tools/Options** command or click **Options** button on the toolbar.  An [Option dialog](#) box will appear.
2. From the **Screen** drop-down list box select Connection Manager item.
3. Correct refresh rate in the **Timer** Interval field. Note that **0** will stop the timer, but you can still use manual refresh after that.
4. Click the **OK** button.

To force screen refresh:

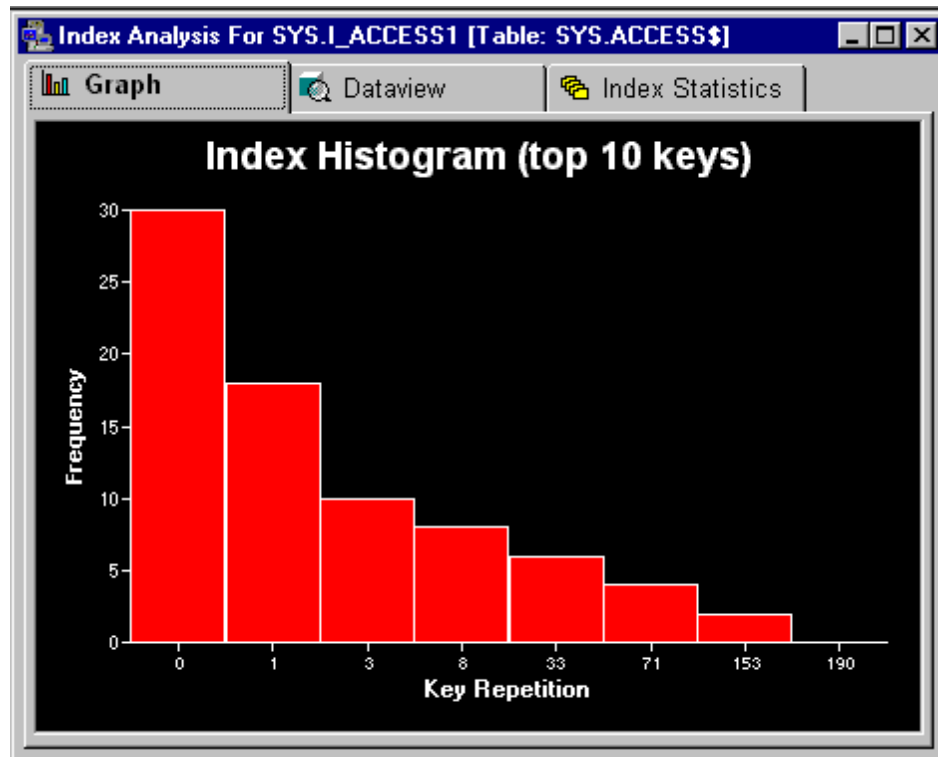
1. Select **File/New Refresh** command (shortcut: CTRL N).

To kill running processes:

1. Highlight the process you want to terminate.
2. Select **File/Delete** command (shortcut: CTRL D) to issue a kill command for that process. You will be prompted with a warning, asking you to confirm the operation.
3. Click the **OK** button if you are sure.

Index Histograms

1. To perform in-depth analysis for an index, select **Tools/Index Analysis (Histogram)** command from application menu. The select index dialog will appear.
2. Select index schema from the **Schema** drop-down list box.
3. Select the required index from **index list**.
4. Click the **OK** button. DB Monitor Expert will issue the analyze command. After database executes the command, a detailed index report will be created.

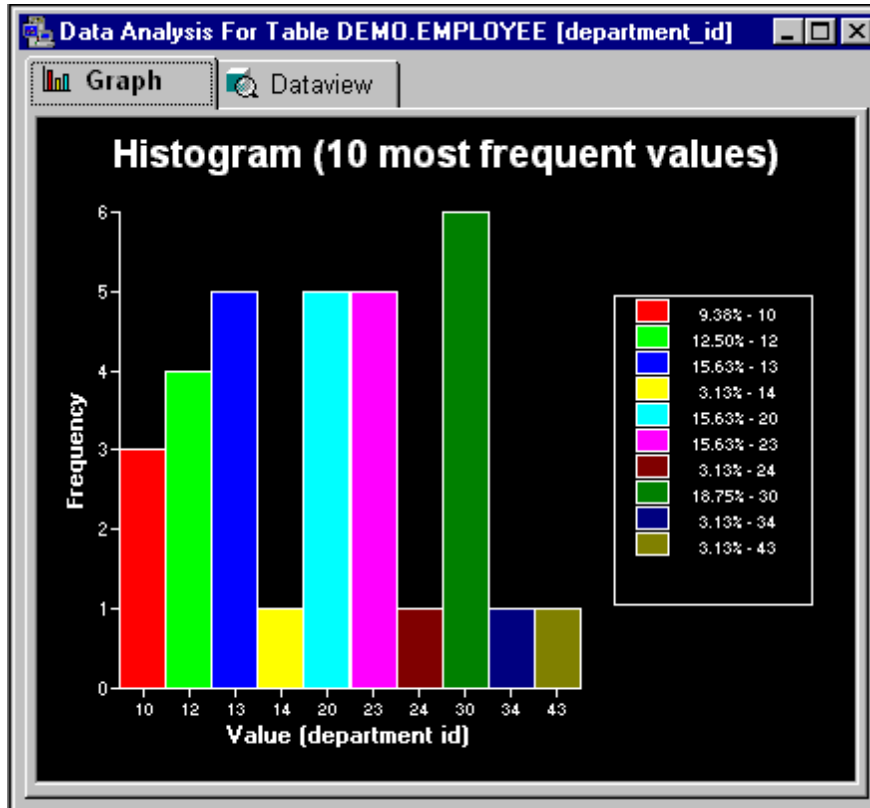


5.

Data Histograms

DB Monitor Expert provides very useful and yet simple tool for quick data analysis. To perform in-depth analysis use DB Monitor Expert facilities to export data into external files (see [Data manipulation and data export](#) topic) then use more sophisticated tools. In case if you have an index on the data that you want to analyze, you can use [Index Histogram](#) tool instead.

1. To perform analysis for a table data select **Tools/Data Analysis (Histogram)** command from application menu. The select table dialog will appear.
2. Select table schema from the **schema** drop-down list box.
3. Select the required table from the **index list**.
4. Select column from which you want to take data for the histogram.
5. Click the **OK** button. DB Monitor Expert will execute all the necessary SQL commands, and creates data histogram report. Note that the time required for data analysis heavily depends on amount of data stored in the column.



Reports

You can use reporting facilities to gather general and detailed information about your database physical and logical structure as well as database security information. We encourage you to use these reports as part of your database documentation. Besides simple printing, you can export report data to numerous file formats as well as select and copy report data to the Windows clipboard for sharing with other applications. See [Working with data and export](#) topic for help on how to export, copy, filter and sort report data. You can also right-click on any report to get some additional features accessible from a popup menu that appears in response to the click.

For most reports, DB Monitor Expert displays [Select Schema dialog](#) before retrieving the report. You can check either **Report All** checkbox for all schemas, or select one or more schemas you want to add to the chosen reports.

Objects Report

Reports summary data for different types of database objects.

Tables Report

Reports detailed information about all database tables. This report queries database repository table - SYS.DBA_TABLES. Oracle updates table statistic in the repository when you [analyze tables](#). To refresh table statistic select **Tools/Table Analyzer** and perform analyzes for tables that haven't been analyzed for a while.

Indexes Report

Reports detailed information about all database indexes. This report queries database repository table - SYS.DBA_INDEXES. Oracle updates table statistic in the repository when you [analyze indexes](#). To refresh

index statistic select **Tools/Index Analyzer** and perform analyzes for indexes that haven't been analyzed for a while. Alternatively, you can analyze tables and/or clusters with cascade indexes option turned on.

Clusters Report

Reports detailed information about all database clusters. This report queries database repository table - SYS.DBA_CLUSTERS. Oracle updates cluster statistic in the repository when you [analyze cluster](#). To refresh cluster statistic select **Tools/Cluster Analyzer** and perform analyzes for clusters that haven't been analyzed for a while.

Tablespaces Report

Reports space allocation information for tablespaces; displays free space in red for tablespaces that have less than 10% of free space. A bar graph is provided for easy comparison of free and allocated space.

Database Files Report

Reports database files, their sizes and statuses; for each file reports tablespaces, it consists of, as well as tablespace sizes and statuses.

Sequences Report

Reports detailed information about all database sequences.

Snapshots Report

Reports detailed information about all database snapshots, including query text for each snapshot.

Synonyms Report

Reports detailed information about all database synonyms.

Database Links Report

Reports detailed information about all database links.

Users Report

Reports detailed information about all database users, including their encrypted passwords.

Profiles Report

Reports detailed information about all database profiles, including associated password security settings and resource usage limits.

Table Privileges Report

Reports all granted table privileges. For your convenience, the report is broken on four groups:

- Group by grantor
- Subgroup by grantee
- Subgroup by scheme
- Subgroup by object

Triggers Report

Reports detailed information about all database triggers, including trigger definitions.

Procedures, Functions And Packages Report

Reports detailed information about all database function and packages, including their code.

Views Report

Reports detailed information about all database views, including view definitions.

Fragmentation Report

Reports all fragmented database objects that have four or more extents. For your convenience, the report data is grouped:

- Group by scheme
- Subgroup by object (segment)

SGA Report

Reports summary information for the System Global Area.

Parameters Report

Reports parameters of the running Oracle instance.



Version Report

Reports versions of the running Oracle server modules.

Custom Reports

In addition to the variety of predefined reports that DB Monitor Expert provides, you can also define your own reports. User-defined reports can address your database specific requirements.


To create a new report:

1. Select **Report/Custom Report** command. DB Monitor Expert will open custom report dialog. The dialog window displays all already created custom reports.
2. Click the **New** button.
3. Enter valid name in the **Name** field on the **Properties** tab. Note that name may consist spaces.
4. Select presentation style from **Style** drop-down list. See [Choosing a presentation style](#) topic for description of available styles.
5. Optionally check **Title**, **Run Date**, and **Page n of m** options
6. Type SQL SELECT statement in the **Select** field. Alternatively, you can paste SQL from the clipboard. To do this, use keyboard shortcut CTRL V or use right mouse click on the **Select** field then from popup menu choose **Paste** command. Also, you can click on the **Open File**  button to paste SQL from the file. This SQL statement will be used to retrieve data for the report you are defining. Make sure that the SQL statement does not end with a semicolon (;). To avoid unclear column names in the report syntax always use aliases in the select statement (see [examples](#)).
7. Click **Filter** button to specify additional client-side [filter](#) a newly created report.
8. Click **Sort** button to change client-side [sort](#) for a newly created report.
9. Click on the **Preview** tab to customize print properties of the report. Click on the  button to bring up [Print Options](#) dialog for more options. For Crosstab reports, you must click on the **Crosstab** button to complete [Crosstab](#) definition of the report.
10. Click on the **Design** tab to customize display properties of the report. Note that you may not

customize display properties for reports that have Crosstab presentation style due to dynamic nature of these reports. The tab page is divided into two parts:

- The "Report visual syntax" pane
- The "Object attributes" pane

You can change the size of the panes. When you make any pane larger, the adjacent pane becomes smaller. To resize panes:

- 1) Move the mouse pointer in between of the panes until you see a black line with two arrows.

- 2) Drag the edge of the pane until the pane is the size you want.

To customize properties of report visual objects:

- 1) Click on the object in the "Report visual syntax" pane. Use pane scrollbars if needed. The object color and background colors get reversed. DB Monitor Expert displays all available properties for the selected object in the "Object attributes" pane
 - 2) Modify required attributes for the object.
11. Click the **Save** button to save the new report definition. The report can be used again later on.
 12. Click on the **Run** button to start the graph you have defined. Note that you can run the graph without doing save first, but by default DB Monitor Expert does not save the graph definition for you until you click the **Save** button. So that if you do not save the definition you will not be able to reuse this graph later.
 13. Now you can build another report or click **Exit** button to close Custom Report dialog.

The following two examples of user-defined reports are included in the standard installation package:

1. Free Space in Tablespace Report

```
select tablespace_name,
       max_blocks,
       count_blocks,
       sum_free_blocks,
       100*sum_free_blocks/sum_alloc_blocks pct_free
from
  (select tablespace_name,
         sum(blocks) sum_alloc_blocks
   from dba_data_files
   group by tablespace_name
  ),
  (select tablespace_name free_space_tablespace_name,
         max(blocks) max_blocks,
         count(blocks) count_blocks,
         sum(blocks) sum_free_blocks
   from dba_free_space
   group by tablespace_name
  )
where tablespace_name = free_space_tablespace_name
```

2. Segment sizes by tablespace, owner and segment type

```
select tablespace_name,
       segment_type,
       owner,
       segment_name,
       sum(blocks) blocks,
       sum(bytes) bytes
from sys.dba_extents
group by tablespace_name, segment_type, owner, segment_name
order by tablespace_name, owner, segment_type
```

See also:

[Monitor connections and resources](#)

[Historical trends - database flow](#)
[Graph - dynamic statistic](#)
[Working with data and data export](#)
[Customization](#)

Choosing a presentation style

Tabular style

The Tabular presentation style presents data with the data columns going across the page and headers above each column. As many rows from the database will display at one time as can fit in the report page.

Form style

The Form style presents data with the data columns going down the page and labels next to each column.

Grid style

The Grid style presents data in a row-and-column format with grid lines separating rows and columns. Everything you place in a grid style report must fit in one cell in the grid. You can reorder and resize columns with the mouse in a grid report during execution. Most Oracle monitor predefined reports have Grid presentation style.

Newspaper style

This style like a Tabular presentation style presents data with the data columns going across the page and headers above each column, then data going across the page in multiple columns, such as in a phone list. For purposes of page fitting, the whole report is considered a single column. You can specify the number of newspaper-style columns in the **Columns** field of the Custom report dialog.

Rich Text style

The Rich Text style presents data with the data columns going down the page and labels next to each column. This style gives you the ability to copy report data during execution to the system Clipboard in the RTF format.

See also: [User-defined Reports](#)

Status Analyzer

Status Analyzer is a simple utility for exploring Oracle database integrity. With a single click, you can find all the database space, objects, and integrity constraints that have one of the following statuses: NOT AVAILABLE, INVALID, OFF LINE, DISABLED. Using that utility, you can get a hierarchical display of such information presented in easy understandable format.

Navigating

To navigate the Status Analyzer use standard tree-view navigation techniques that you use when working with Windows Explorer and other applications. For quick expanding or collapsing of multiple items use right mouse button clicks to invoke the popup menu. Use the following commands available in the popup menu to quickly expand or collapse multiple items:

- **Expand Branch**
- **Collapse Branch**
- **Expand All**
- **Collapse All**

Which items are displayed?

Status Analyzer displays only context sensitive items. For example if there is no off line rollback segments, the Status Analyzer will not show the **Rollback Segments** category.

Compile Invalid Objects

You can this tool easily recompile individual objects, group of invalid objects or all invalid objects in the database. Select any invalid objects and choose the appropriate action from the **File/Recompile** menu. The Status analyzer view is automatically refreshed after the re-compilation is complete.

Zooming

You can use zoom feature (select **View/Zoom** command) to zoom in/out security information.

Printing

You can print information displayed in the Status Analyzer panes by choosing File/Print command (shortcut: CTRL P). You can preview how it will look when printed by choosing File/Print Preview. See: [Print](#) topic for information on print and print preview options.

Dependencies Analyzer

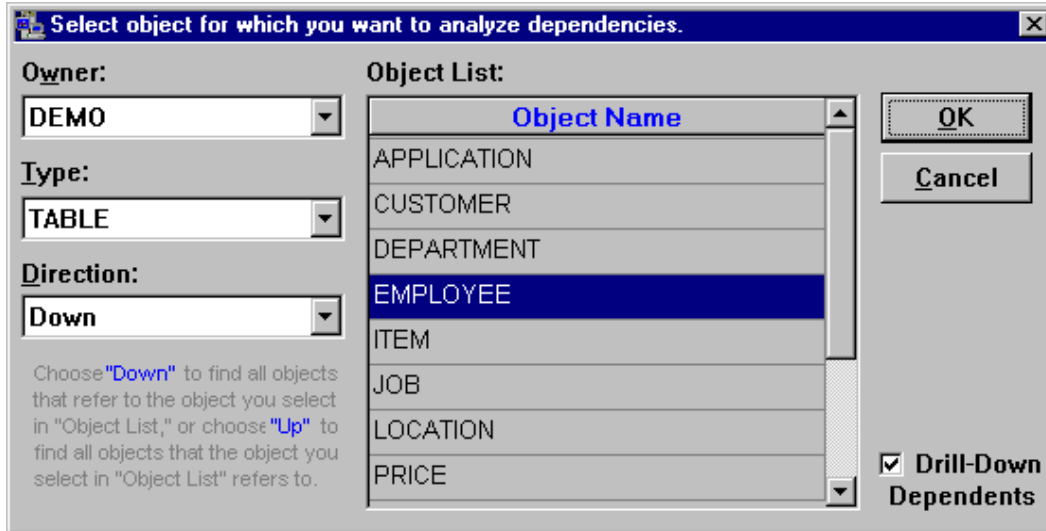
The definitions of certain objects, such as views and procedures, reference other objects, such as tables. Therefore, some objects are dependent on the objects referenced in their definitions. An object that references another object as part of its definition is called a dependent object, while the object being referenced is a referenced object. Oracle defines the following types of dependent and referenced objects.

Dependent Objects	Referenced Objects
View	Table
Procedure	View
Function	Sequence
Package Specification	Synonym
Package Body	Procedure
Type	Function
Type Body	Package Specification
Trigger	Type specification
	Operator

In addition to types specified in the table above, Dependencies Analyzer tracks logical dependencies such as data references indicated by foreign keys.

Dependencies Analyzer creates multi-level hierarchical display of both physical and logical dependencies information presented in easy understandable format.

To run Dependencies Analyzer select **Tools** menu then **Dependencies Analyzer** command. DB Monitor Expert will display **Select Object** dialog box. You use that dialog to select root object for which you want to find dependents or referenced objects.



1. Select object schema from the **Owner** drop-down list.
2. Select object type from the **Type** drop-down list. DB Monitor Expert will populate list of objects of specified type for specified schema.
3. Select object name from the **Object list**.
4. Select what kind of references you are looking for in the **Direction** drop-down list. Choose "Down" to find all objects (dependents) that refer to the object you select in the **Object List** or choose "Up" to find all objects that the selected object refers to.
5. Check **Drill-Down Dependents** checkbox if you want Dependencies Analyzer automatically find and expand dependents for dependents. Note that in that case existence of foreign key for self-referenced tables may cause a dead loop.

Navigating

To navigate the Dependencies Analyzer use standard tree-view navigation techniques that you use when working with Windows Explorer and other applications. For quick expanding or collapsing of multiple items use right mouse button clicks to invoke the popup menu. Use the following commands available in the popup menu to quickly expand or collapse multiple items:

- **Expand Branch**
- **Collapse Branch**
- **Expand All**
- **Collapse All**

Zooming

You can use zoom feature (select **View/Zoom** command) to zoom in/out security information.

Printing

You can print information displayed in the Dependencies Analyzer panes by choosing **File/Print** command (shortcut: CTRL P). You can preview how it will look when printed by choosing **File/Print Preview**. See: [Print](#) topic for information on print and print preview options.

DB Trends Expert

The DB Trends package is designed to simplify and automate tedious database monitoring and maintenance tasks that every Oracle DBA has to perform everyday.

Data captured by DB Trends can be used for proactive database monitoring and capacity planning. DB Trends is tightly integrated with [DB Monitor Expert](#) and [DB Tuning Expert](#). Other components like [DB Space Expert](#) also use the collected data for predicting various database problems before they occur.

The package definition can be found in the [TRENDS_INSTALL.SQL](#) file located in the DB Monitor Expert Installation directory. This file includes complete description of DB Trends business functions and usage.

See [Installation/Uninstallation](#) topic for instructions on installing/uninstalling DB Trends.

DB Trends - Installation/Uninstallation

DB Trends package can be installed using graphical [Database Setup utility](#) provided with the DB Tools. DB Trends can be also installed using the [TRENDS_INSTALL.SQL](#) script that can be found in DB Tools installation directory

To create this package using graphical Database Setup utility launch the utility and answer several questions asked by the Setup Wizard.

To create the package manually:

1. Start DB Monitor Expert.
2. Connect to database as SYS.
3. Open [TRENDS_INSTALL.SQL](#) file in the DBA Notepad
4. Use interface variables that can be found in the beginning of the file to customize package behavior. See comments in the script for additional information.
5. Execute the script.
6. Use the [Trends Selection](#) tool (**Trends/Trends Selection** menu) to select/deselect objects and spaces that you want to include for the automated analysis and trend data collection process.



Tip: It is highly recommended to schedule [DB_TRENDS.COLLECT](#) procedure to run on a regular basis. You may want to use the standard [DBMS_JOB](#) package for this task. Use DB Monitor Expert **Trends** tools to analyze, display, and graph the collected trend data.

Upgrading DB Trends package

If you have previously installed DB Trends version 1.x you need to run [TRENDUPG.SQL](#) script to upgrade the package to version 2.0. This script file can be found in the DB Monitor Expert Installation directory.


Disabling DB Trends data collections

Use DB Job Expert to disable DB Trends Data Collection job

Uninstalling DB Trends package

1. Drop the following tables:
ORA_MONITOR.FLOW_TABLE

- ORA_MONITOR.OBJECT_WATCH
- ORA_MONITOR.INSTANCE_WATCH
- 2. Drop ORA_MONITOR.DB_TRENDS package

 **Important Note:** The DB Trends installation script creates ORA_MONITOR user identified by IMONITOR password. The [DB Difference Expert](#), [DB Monitor Expert](#), and [DB Audit Expert](#) tools also create repository objects in the ORA_MONITOR schema. You should drop this schema only in case if you are completely uninstalling/removing DB Tools from your system.

Historical Trends

DB Trends Expert can collect and store hundreds different statistics for historical trends analysis. Most statistics are stored in the FLOW_TABLE. We encourage you to use this repository table for your own reporting.

When you select any **Trends** command from the DB Monitor Expert **Trends** menu for a first time, DB Monitor Expert will create the ORA_MONITOR schema (if it does not exist yet) and the FLOW_TABLE in that schema. You must have CREATE TABLE privilege in order to continue. The FLOW_TABLE table serves as a repository table in which both DB Monitor Expert and DB Trends Expert save trend data.

Every time when you open any trend report DB Monitor Expert, a new data collection is performed by default. Newly collected data is added to the repository table. All users, who will use trend reports and graphs, must have INSERT/UPDATE privilege for the FLOW_TABLE.

If for any object or group of related objects, the trend data collection is performed first time the trends graph is unavailable, because of insufficient information yet to build the trends graph.

For your convenience, all trend reports are grouped by categories such as Table, Index, etc...

Notes:

- Remember that all trend reports heavily depend on data in the Oracle system catalog tables. Oracle updates these tables when you execute [table, index, and cluster analyze](#) commands.
- Analyze database object often to get precise trends reports. See the following help sections for instructions on collecting and graphing database trends.

Table Trends

1. Select **Trends/Table Trends** command. A table trends dialog will appear.
2. Select schema from the **schema** drop-down list box
3. Select table to be reported from the **table list**. Note that active selection has blue background for the entire row. To add more tables, hold down CTRL key and click on tables you want to add to the batch. To select a range of tables, hold down SHIFT key and click on the last table that you want to add. All tables between the last selected and the clicked one will be highlighted.
4. Uncheck **Update** box to disable collection of the new information for the report.
5. You may want to limit new report in time. Enter start and end dates in corresponding fields to specify trends interval. To disable time restrictions and create new report for the all-available data stored in the repository, check **No limit** box.
6. Press **OK** button. DB Monitor Expert will start the new data collection. After completion of the new collection, trend reports for the last selected table will be created.

Index Trends

1. Select **Trends/Index Trends** command. An index trends dialog will appear.
2. Select schema from the **schema** drop-down list box
3. Select index to be reported from the **index list**. Note that active selection has blue background for the entire row. To add more indexes, hold down CTRL key and click on indexes you want to add to the batch. To select a range of indexes, hold down SHIFT key and click on the last index that you want to add. All indexes between the last selected and the clicked one will be highlighted.
4. Uncheck **Update** box to disable collection of the new information for the report.
5. You may want to limit new report in time. Enter start and end dates in corresponding fields to specify trends interval. To disable time restrictions and create new report for the all-available data stored in the repository, check **No limit** box.
6. Press **OK** button. DB Monitor Expert will start new data collection. After completion of the new collection, trend reports for last selected index will be created.

Cluster Trends

1. Select **Trends/Cluster Trends** command. A cluster trends dialog will appear.
2. Select schema from the **schema** drop-down list box
3. Select cluster to be reported from the **cluster list**. Note that active selection has blue background for the entire row. To add more clusters, hold down CTRL key and click on clusters you want to add to the batch. To select a range of clusters, hold down SHIFT key and click on the last cluster that you want to add. All clusters between the last selected and the clicked one will be highlighted.
4. Uncheck **Update** box to disable collection of the new information for the report.
5. You may want to limit new report in time. Enter start and end dates in corresponding fields to specify trends interval. To disable time restrictions and create new report for the all-available data stored in the repository, check **No limit** box.
6. To include in batch all cluster tables and indexes, check **Cascade Tables & Indexes** box.
7. Press **OK** button. DB Monitor Expert will start new data collection. After completion of the new collection, trend reports for last selected cluster will be created.

Schema Trends

1. Select **Trends/Schema Trends** command. A schema trends dialog will appear.
2. Select schema from the **schema** drop-down list box
3. Uncheck **Update** box to disable collection of the new information for the report.
4. You may want to limit new report in time. Enter start and end dates in corresponding fields to specify trends interval. To disable time restrictions and create new report for the all-available data stored in the repository, check **No limit** box.
5. To include in batch all tables and indexes for that schema check **Cascade Tables & Indexes** box.
6. Press **OK** button. DB Monitor Expert will start new data collection. After completion of the new collection, trend reports for the chosen schema will be created.

Tablespace Trends

1. Select **Trends/Tablespace Trends** command. A tablespace trends dialog will appear.
2. **Select the tablespace** to be reported from the **tablespace list**. Note that active selection has blue background for the entire row. To add more tablespaces, hold down CTRL key and click on tablespaces you want to add to the batch. To select a range of tablespaces, hold down SHIFT key

and click on the last tablespace that you want to add. All tablespaces between the last selected and the clicked one will be highlighted.

3. Uncheck **Update** box to disable collection of the new information for the report.
4. You may want to limit new report in time. Enter start and end dates in corresponding fields to specify trends interval. To disable time restrictions and create new report for the all-available data stored in the repository, check **No limit** box.
5. To include in batch all tables and indexes that are placed in selected tablespaces, check **Cascade Tables & Indexes** box.
6. Press **OK** button. DB Monitor Expert will start new data collection. After completion of the new collection, trend reports for last selected tablespace will be created.

File Trends

1. Select **Trends/File Trends** command. A file trends dialog will appear.
2. Select file to be reported from the **tablespace list**. Note that active selection has blue background for the entire row. To add more files, hold down CTRL key and click on files you want to add to the batch. To select a range of files, hold down SHIFT key and click on the last file that you want to add. All files between the last selected and the clicked one will be highlighted.
3. Uncheck **Update** box to disable collection of the new information for the report.
4. You may want to limit new report in time. Enter start and end dates in corresponding fields to specify trends interval. To disable time restrictions and create new report for the all-available data stored in the repository, check **No limit** box.
5. To include in batch all tables and indexes that are placed in selected files, check **Cascade Tables & Indexes** box.
6. Press **OK** button. DB Monitor Expert will start new data collection. After completion of the new collection, trend reports for last selected file will be created.

Rollback Segment Trends

1. Select **Trends/Rollback Segments Trends** command. A rollback segment trends dialog will appear.
2. Select rollback segment to be reported from the **rollback segments list**. Note that active selection has blue background for the entire row. To add more segments, hold down CTRL key and click on segments you want to add to the batch. To select a range of segments, hold down SHIFT key and click on the last segment that you want to add. All segments between the last selected and the clicked one will be highlighted.
3. Uncheck **Update** box to disable collection of the new information for the report.
4. You may want to limit new report in time. Enter start and end dates in corresponding fields to specify trends interval. To disable time restrictions and create new report for the all-available data stored in the repository, check **No limit** box.
5. Press **OK** button. DB Monitor Expert will start new data collection. After completion of the new collection, trend reports for last selected rollback segment will be created.

Database Trends


1. Select **Trends/Database Trends** command. A database trends dialog will appear.
2. Uncheck **Update** box to disable collection of the new information for the report.
3. You may want to limit new report in time. Enter start and end dates in corresponding fields to specify trends interval. To disable time restrictions and create new report for the all-available data stored in the repository, check **No limit** box.

4. To include in batch all database tables and indexes, check **Cascade Tables & Indexes** box.
5. Press **OK** button. DB Monitor Expert will start new data collection. After completion of the new collection, trend reports for the database will be created.

Custom Trend Graph

In addition to the variety of predefined trend graphs that DB Monitor Expert provides, you can also define your own graphs. User-defined trend graphs can address your database specific requirements. Using custom graphs, you can also automate problem detection and correction.

To create a new trend graph:

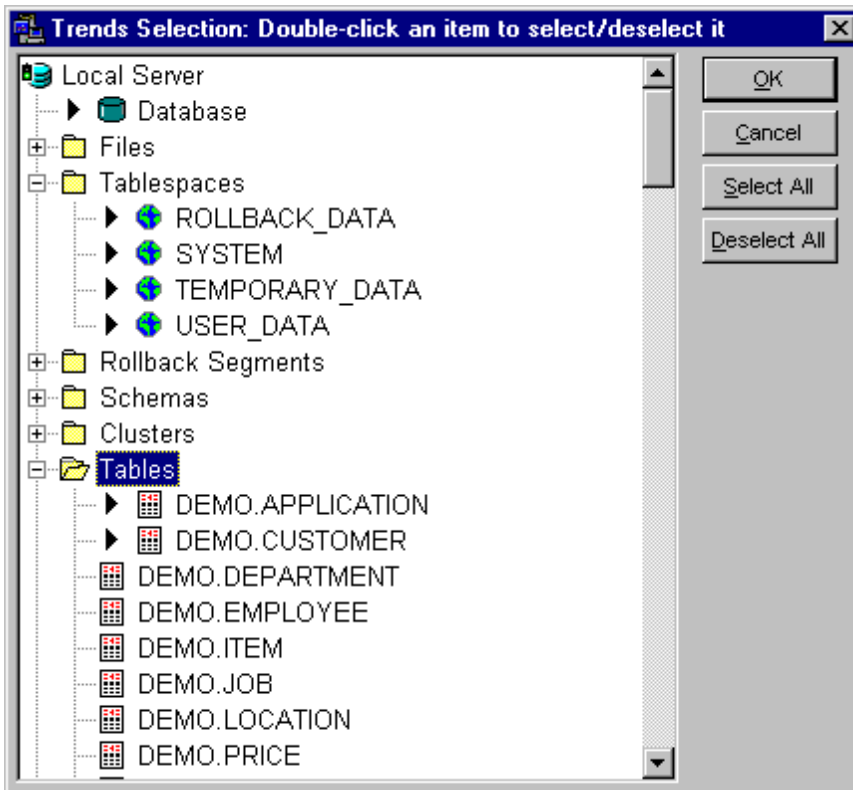
1. Select **Trends/Custom Trend Graph** command. DB Monitor Expert will open custom trend graph dialog. The dialog window displays all already created custom trend graphs.
2. Click the **New** button.
3. Enter valid name in the **Name** field on the **Properties** tab. Note that name may consist spaces.
4. Type SQL SELECT statement in the **Select** field. Alternatively, you can paste SQL from the clipboard. To do this, use keyboard shortcut CTRL V or use right mouse click on the **Select** field then from popup menu choose **Paste** command. Also, you can click on the **Open File**  button to paste SQL from the file. This SQL statement will be used to gather statistics to be displayed in the monitor you are defining. Make sure that the SQL statement does not end with a semicolon (;). To avoid unclear column names in the graph syntax always use aliases in the select statement (see [examples](#)). **Important:** make sure to include **timestamp** column in the SELECT list. DB Monitor Expert uses this column for the trend graph X-axis.
5. Click on the **Preview** tab to customize display properties of the graph.
6. Select column from the **Value** drop-down list. That column will be graphed.
7. If you want to display multiple graphs (series of characteristics) simultaneously, use the **Series** drop-down list to select column that will be used for series. For display of single value leave **Series** field blank or select **<None>** from the list.
8. Click **Save** button to save the new trend graph definition. The graph can be used again later on.
9. Click **Run** button to start the graph you have defined. Note that you can run the graph without doing save first, but by default DB Monitor Expert does not save the graph definition for you until you click the **Save** button. So that if you do not save the definition you will not be able to reuse this graph later.
10. Now you can build another trend graph or click **Exit** button to close Custom Trend Graph dialog.

Notes:

- You can clean up the repository table if it stores too many records for the reported period (the report takes too long to complete). To remove unnecessary data select the **Dataview** tab on the trends report, highlight rows you want to remove and then elect the **File/Delete** command (shortcut: CTRL D). Select **File/Save** command (shortcut: CTRL S) to save changes.
- A data collection and report generation time depend on the three following factors:
 1. Database load (busy rate)
 2. Number of objects/spaces selected for the new data collection
 3. Volume of trend data available for the selected object

Selecting Objects and Spaces for Automated Trend Data Collections

If you are using or planning to use the [IDH_DBTRENDS](#) package supplied with the DB Monitor Expert you can use **Trends/Trends Selection** menu in DB Monitor Expert component to select objects and spaces for automated trend data collections. You must login as ORA_MONITOR (or use matching login in case if you have customized the DB_TREND package) in order to use the **Trends Selection** tool. The **Trends Selection** dialog presents graphically information stored in the ORA_MONITOR.OBJECT_WACTH table. See comments in [TRENDS_INSTALL.SQL](#) file for detailed table description.



After the **Trends Selection** dialog appears:

- 1 Expand the required section
- 2 Double-click on the required objects and spaces to submit them for automated monitoring and trend data collections. Note that the second double-click on the previously select object or space deselects it and removes from the list of objects and spaces to be processed by the DB Trends package. To select all objects/spaces in a section, highlight the required section then click **Select All** button. To deselect all objects/spaces in a section, highlight the required section then click **Deselect All** button.
- 3 Click **OK** button.



Note: Selected objects and spaces feature ► symbol in front of their icons.

DB Difference Expert

Overview

You can use DB Difference Expert to compare entire Oracle databases as well as database schemas and even individual objects. Compared databases, schemas, or objects can reside on one or different Oracle servers. DB Difference Expert creates and uses the central repository tables for definitions of compared items. Whenever you compare databases, schemas or objects, DB Difference Expert adds their definition to the repository. That's why DB Difference Expert allows also tracking object and database changes over a period of time. It is also capable to compare different versions of objects previously saved in the repository.

Before You Begin

You must configure the DB Difference Expert repository before you start using DB Difference Expert. When you start DB Difference Expert first time, it prompts to create the repository. You use the Repository Options dialog to customize the repository. You can drop or update the repository at any time by selecting **File/Drop Repository** and **File/Update Repository** commands from the DB Difference Expert menu. If you drop the repository and attempt to start new comparison, DB Difference Expert will prompt you to recreate the repository. This way you can move the repository to a different Oracle server if there is a need to do so.

DB Difference Expert also provides you with graphical repository browser to that you can use to manually browse and clean up the repository similar to how you use File Explorer to browse files on your computer. To open the repository Browser, select **File/Browse Repository** command.

The repository must be created in the ORA_MONITOR schema. If this schema does not exist DB Difference Expert attempts to create it at the time of repository creation. Good repository configuration is one of the important factors contributing to the performance of various comparison operations. It is highly recommended, that you choose different devices when specifying tablespaces for the repository tables and indexes.

You must have CREATE TABLE, CREATE USER privileges in order to create the DB Difference Expert repository.

You must have SELECT TABLE privilege on the system catalog tables to be able to perform schema and object comparison. You must have SELECT ANY TABLE privileges to be able to compare entire databases.

If you do not have sufficient authority to use DB Difference Expert, contact your database administrator.

Comparison Options

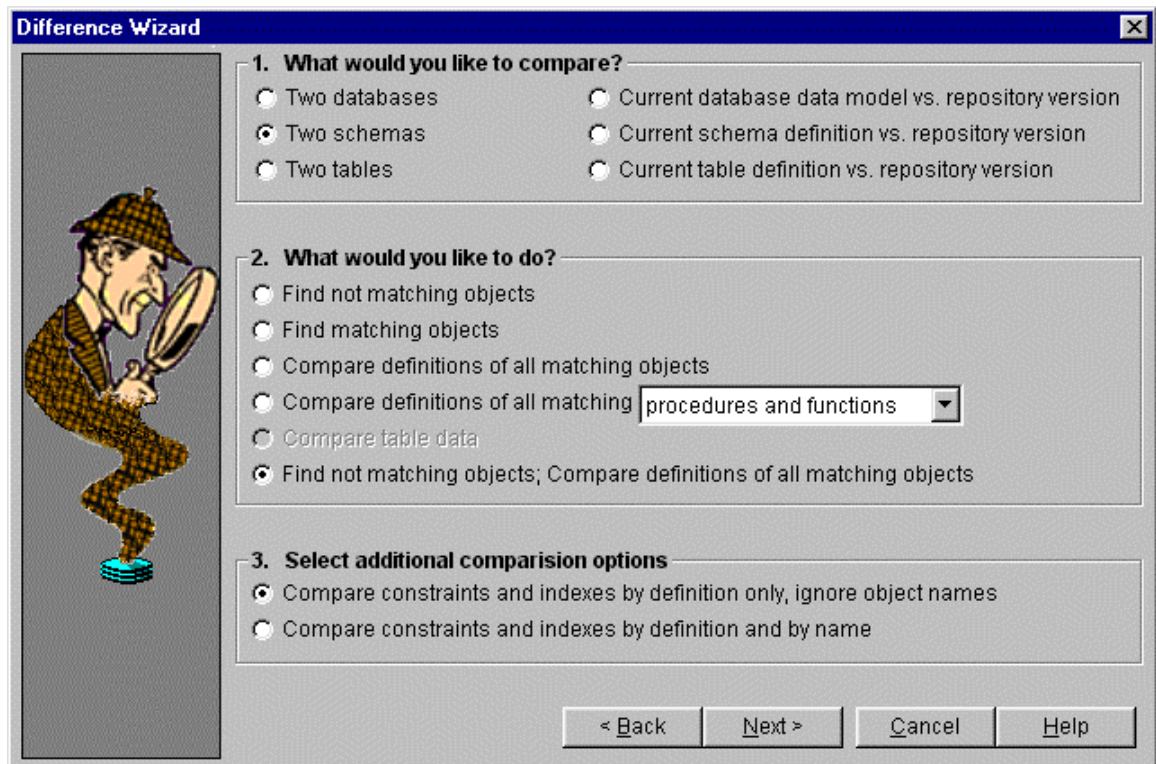
When to use and when not to use the "Compare constraints and indexes by definition only, ignore object names" option?

By default, Oracle generates unique names for table indexes and constraints in SYS_XXXXX format unless you specifically supply the required names. In most cases, the actual name is not important so when compare two tables you are interested in verifying that both tables have the same index by definition. Therefore, **"Compare constraints and indexes by definition only, ignore object names"** option instructs DB Difference Expert to ignore their names when comparing one or more tables and/or clusters.

If in your application you are referencing particular indexes and constraints by name, then you should choose **"Compare constraints and indexes by definition only and by object names"** instead. This ensures that DB Difference Expert will report as identical only these objects that exactly match each other including their names.

How to Analyze

1. To start new difference analysis, select **File/New Analysis** command from the DB Difference Expert menu. The Difference Wizard will appear.



2. Select analysis scope (one of the following):
 - Two databases
 - Two schemas
 - Two tables
 - Current database data model vs. repository version
 - Current schema definition vs. repository version
 - Current table definition vs. repository version
 - Two code objects (source code comparison)
3. Select appropriate items for comparisons. Note that DB Difference Expert enables only these items that are valid for the previously selected analysis scope.
4. Select additional comparison options:
 - Compare constraints and indexes by definition only, ignore object names
 - Compare constraints and indexes by definition only and by object names
5. Click the Next button
6. Specify database connection information then click the Next button again
7. If you are analyzing schemas or table select the required schema names from the drop-down lists
8. If you are analyzing tables, also select the required table names from the drop-down lists.
9. Click the Next button, review the analysis execution plan and if satisfied click the Start button.

Understanding and Using Comparison Results

The DB Difference Expert provides you with three different views for the analysis results:

- Tree-view style expandable/collapsible difference navigators
- SQL scripts
- Comparison Report

When comparing large number of objects the difference navigators are by far the most effective and easily readable presentation style for presenting comparison results. Note that "different" objects are displayed in bold. The difference navigators provide you with the easy-to-use drill-down capabilities.

You can use the generated SQL scripts as templates in case if you need to synchronize the compared items. You should always verify these scripts. Because the generated SQL scripts are not guaranteed to be error-free, you execute them at your own risk. SoftTree Technologies does not guarantee 100% accuracy. These SQL scripts are intended to help you avoid time-consuming tasks as the database such as reverse-engineering.

Difference navigator trees and SQL scripts are optimized for the screen. If you wish you can print them using **File/Print** command. Please keep in mind that the Comparison report is primarily optimized for the printing and should be accordingly.

DB Index Expert

Overview

How DB Index Expert Can Help

DB Index Expert can help DBAs and developers to optimize and monitor existing databases. In general, it can help in answering the following questions:

- **What are new indexes that can improve your application/database performance?**
- **Are there unneeded indexes that cause your application to run slowly?**
- **What are optimal physical and logical parameters for a specific index?**
- **Is it the time when a specific index will benefit from being rebuilt?**

In addition, DB Index Expert can help you automate some routine database maintenance tasks such as periodical index and table statistical analyses that help Oracle optimizer to choose best execution plans for your database queries.

Before You Begin

You must have SELECT privileges on the system catalog tables in order to use DB Index Expert. In addition, you need access to Oracle dynamic performance tables. If you do not have sufficient authority to use DB Index Expert, ask your database administrator to grant you 'SELECT ANY TABLE' privileges or grant you access to following tables: ALL_TAB_COLUMNS, ALL_IND_COLUMNS, ALL_CONS_COLUMNS, ALL_INDEXES, ALL_TABLES, ALL_CONSTRAINTS, SYS.INDEX_STATS, SYS.INDEX_HISTOGRAM, SYS.DBA_EXTENTS, SYS.DBA_SEGMENTS, V\$SQLAREA, V\$VERSION.

It is also highly recommended to have the ANALYZE ANY privilege so you can experience all the features of the DB Index Expert.

Oracle8 users may need grants to use DBMS_SQL package.

Command Line Parameters

Refer to the following table for command line parameters supported by DB Index Expert:

Parameter	Description	Example
/?	Display command line parameters help screen	/?
/O	Connection Driver name. Valid names are: ODBC, O73, O84, O90	/ODBC
/C	Oracle server connect string that to be used to log on to the server	/Coraserv
/U	The user ID that to be used to log on to the server	/Uscott
/P	The password to be used to log on to the server	/Ptiger
/V	Oracle version identifier	/V7.2
/S	Name of the database schema to analyze. If this option is not specified then all schemas will be analyzed	/Sscott
/I	Name of the index to analyze. Use short index names without schema name prefixes. Use /S option to specify schema names.	/lindx_employee_ssn
/A	Turn off printing of generated analysis reports	/A
/H	Output report to HTML file. The file "index.htm" will be created in the DB Index Expert installation directory. This option does not affect report printing. To suppress the printing use option /A	/H
/R	Output report to RTF file. The file "index.rtf" will be created	/R

	in the DB Index Expert installation directory. This option does not affect report printing. To suppress the printing use option /A	
/F	<p>Name of the parameters file. This file must be a standard flat text INI. The file should have at least one section named "Options"</p> <p>Following INI file keys can be used as substitutes for command line parameters:</p> <pre>DBMS - /O Server - /C UID - /U Password - /P Version - /V Schema - /S Index - /I Print - /A Text - /R HTML - /H</pre> <p>The /F parameter can be specified along with other command line parameter. DB Index Expert processes parameter file like if the options specified in the file were specified on the command line.</p>	<pre>[Options] DBMS=O84 Server=OraServ UID=SCOTT Password=TIGGER Schema=INVENTORY</pre>

Example:

[db_tools.exe /index /Oo84 /Cmyserver /Uscott /Ptigger /Sscott](#)

This line will start DB Index Expert in command line mode and instruct it to connect to the Oracle 8 database using **myserver** host string as user **scott** using **tiger** password. DB Index Expert will analyze all indexes and tables in the **scott** schema.

Analysis Options

Refer to the following 2 tables for analysis options:

Analysis Options for Indexes

Options	Description
Analyze and Validate Index Structure	<p>Send "ANALYZE INDEX" command to the database. Command parameters depend on the options chosen using "Analysis Method" dialog:</p> <ul style="list-style-type: none"> • Compute statistics for all rows • Estimate statistics for first <n> rows • Estimate statistics for first <m> % of rows • Select best method based on the object size - this instructs DB Index Expert to automatically choose the method using the following rule: if an index size is less than 1MB then use "Compute statistics for all rows", otherwise use "Estimate statistics for first <m> % of rows" where <m> is calculated as percent of 1MB compared to the index size but not less than 1%.
Analyze Indexed Table	<p>Send "ANALYZE TABLE" command to the database. The command is issued for the table the analyzed index belongs to. If this option is not specified, DB Index Expert counts rows in the table and queries SYS.DBA_SEGMENTS and SYS.DBA_EXTENTS tables as opposite to retrieving information from ALL_TABLES table after table analysis is completed. Note that DB Index Expert analyzes a table only once for all table indexes.</p>

	<p>Command parameters depend on the options chosen using "Analysis Method" dialog:</p> <ul style="list-style-type: none"> • Compute statistics for all rows • Estimate statistics for first <n> rows • Estimate statistics for first <m> % of rows • Select best method based on the object size - this instructs DB Index Expert to automatically choose the method using the following rule: if a table size is less than 1MB then use "Compute statistics for all rows", otherwise use "Estimate statistics for first <m> % of rows" where <m> is calculated as percent of 1MB compared to the table size but not less than 1%.
Validate Index Placement	Check proper tablespace separation for the index and the indexed table.
Validate Table Size	Identify small lookup tables. An index is not efficient when full table scan takes less number of I/Os operations than number of I/Os needed for accessing index root node and retrieving the requested data.
Analyze Index Cardinality	Analyze index keys cardinality (count distinct keys). Create index histogram (non-unique indexes only). Validate that index type is properly chosen.
Analyze Index Balance	Analyze index splitting and spawning and detect out-of-balance indexes. Splitting occurs when an index node is filled with the keys and a new node is created at the same level as the full node. Splitting widens a b-tree horizontally. Spawning occurs when a new level is added to an index. A newly created index is one level index. As keys are added, a spawning takes place, and the first-level node reconfigures itself to have pointers to lower-level nodes. An index may become out-of-balance after a large number of insert and delete operations.
Analyze Index Columns	Analyze index columns and verify proper order of columns in the key. Check efficiency of the column data types.
Analyze Data Distribution	Count distinct values in all index columns. Build data distribution histogram for each column.
Analyze Clustering Factor	Analyze clustering factor and index balance.
Analyze Index Efficiency	Analyze overall index efficiency.
Search Non-unique Index for Uniqueness	Verify index type, search non-unique indexes that can be converted to unique. A unique index usually provides better performance may be achieved as well as some additional index space recovered comparing to non-unique index.
Check Non-unique Index for Uniqueness	Verify index type, search non-unique indexes that can be converted to unique. A unique index usually provides better performance may be achieved as well as some additional index space recovered comparing to non-unique index.
Search Intersecting Indexes	Search indexes that share all its columns with other indexes. Some of these indexes may be dropped as unnecessary. This will improve performance of insert, delete, update operations and recover additional index space.
Analyze Index Space Fragmentation	Analyze space fragmentation.

Analysis Options for Schemas

Options	Description
Search Referential Constraints for Missing Indexes	Search foreign keys that do not have an index created on the constraint columns.
Analyze SQL from Library Cache for Full Table Scans and Used Indexes	Analyze SQL from database Library Cache and use embedded EXPLAIN PLAN to detect performed full table scans and find out indexes being used for the SQL.

Report offending SQL statements.

How to Analyze

Interactive mode

DB Index Expert has easy-to-use standard "wizard" style interface. Perform following steps to use DB Index Expert in interactive mode.

- Start DB Index Expert by double clicking on the Analyzer icon in the DB Tools program group.
- Choose required [analysis options](#). Click "Next" button.
- Select additional analysis and report generation options. Click "Next" button.
- Enter connect information required for establishing a new database connection. Click "Next" button again.
- Select one or more schemas to be analyzed. You can use "Select All", "Deselect All" buttons to quickly select / deselect all listed schemas. Click "Next" button.
- Select one or more indexes to be analyzed. You can use "Select All", "Deselect All" buttons to quickly select / deselect all listed indexes. Click "Next" button. DB Index Expert will process all chosen indexes and generate analysis reports. If needed, click "Stop" button to interrupt the process.
- After reviewing and/or printing generated reports close Analysis Reports window and click "Done" button on the DB Index Expert window to exit or click "Back" button to choose new options and/or schemas/indexes for analysis.

Note: you can click "Back" button at any time to return to the previous options dialog.

Batch mode

DB Index Expert provides alternative command line interface that makes it suitable for unattended batch mode jobs. For instance, a large index can take a long time to analyze and a DBA can carefully choose the right time for analysis and schedule this task.

Read [Command Line Parameters](#) topic for details.

Reports

Depending on the chosen analysis options DB Index Expert can generate 3 different report types whose description follows.

Index Analysis Report

Index Analysis Report is a detail report generated for each analyzed index. This reports physical and logical index parameters, including detailed information on indexed columns and table. The reports may include index improvement suggestions generated by embedded Intelligent Expert Module.

Referential Constraints Analysis Report

This report finds displays all analyzed foreign keys that do not have matching indexes.

Library Cache Analysis Reports

Full Table Scans Report

This report displays tables that caused Full Table Scans. The report also includes summary execution characteristics for SQL statements causing full table scans for these tables. Use this report to pinpoint missing indexes.

Used Indexes Report

This report displays indexes that were by SQL queries found in the database library cache. The report also includes summary of execution characteristics for SQL statements using these indexes. Use this report to verify that Oracle optimizer chooses indexes as you expected. This report also can help you pinpoint unneeded indexes – ones that you don't want Oracle to use in some cases.

Detailed Library Cache Analysis Report

This report provides detailed information on SQL statements for both Full Table Scans and Used Indexes reports.

Report Formats

All reports can be generated in the following formats:

- Print Preview
- Enhanced hypertext mark-up language (HTML) file format. You need a Web browser that supports frames in order to view and print reports. We recommend Netscape Navigator 3.x and above or Microsoft Internet Explorer 3.x and above. The reports are created as a set of HTML files in the DB Index Expert installation directory. After report generation, DB Index Expert automatically launches your default Web browser.
- ASCII text-format file that can be read in Windows Notepad and any other word-processors. The report is created in the DB Index Expert installation directory.

DB Tuning Expert

Overview

The Oracle Server is highly tunable. You can enhance database performance by adjusting database applications, the database itself, and the operating system. Proper tuning of Oracle provides the best possible database performance for your specific application and hardware configuration.

What DB Tuning Expert Can Do For You

DB Tuning Expert is a great aid for those who are responsible for the operation, maintenance, and performance of an Oracle Server. If you are one of them then DB Tuning Expert can help you identify quickly many database performance problems. With a few mouse clicks can run robust database performance and organization diagnostic tests and get a comprehensive report on the existing performance problems along with the expert recommendations for solving all found problems and avoiding them in the future. DB Tuning Expert can analyze databases in the following areas:

- **Configuration and statistics for Oracle database instance; database physical and logical models.**
- **Various statistics for schema objects, including fragmentation, design, placement and more...**
- **Performance of SQL statements from the Library Cache.**
- **Database trends using data collected by the DB Trends package.**

Before You Begin

You must have SELECT privileges on the system catalog tables and dynamic performance views. If you do not have sufficient authority to use DB Tuning Expert, ask your database administrator to grant you 'SELECT ANY TABLE'.

Oracle8 users may need grants to use DBMS_SQL package.

How To Tune Your Database

The Oracle Server is a highly tunable software product. Its flexibility allows you to make small adjustments that affect database performance. By tuning your system, you can tailor its performance to best meet your needs. However, tuning is usually a series of trade-offs. Once you have determined the bottlenecks, you may have to sacrifice some other areas to achieve the required results. For example, if I/O is a problem, you may need to purchase more memory or more disks. If a purchase is not possible, you may have to limit the concurrency of the system to achieve the required performance. If you have clearly defined goals for performance, the decision on what to trade for higher performance is simpler because you have identified the most important areas.

DB Tuning Expert findings and recommendations can help you determined the current bottlenecks, however they cannot be 100% accurate because of the following reasons:

- DB Tuning Expert does not have much historical data and therefore does not analyze the history of the database performance over time
- DB Tuning Expert does not (and has no way) analyze the performance of the Operation System and the hardware on the machine where your Oracle Server resides
- DB Tuning Expert recommendations are not guaranteed to match your performance goals
- DB Tuning Expert does not attempt to generate balanced solutions, it simply reports found problematic areas

Still, DB Tuning Expert can provide you with a comprehensive performance analysis reports that you can use in tuning database on day-to-day basis. Only everyday performance monitoring and tuning helps you to maintain a well-tuned system. Keeping a history of the database performance over time enables you to make useful comparisons. By comparing multiple resource consumption reports generated by DB Tuning Expert over a period of time, you can conduct objective scalability studies and from these predict the resource requirements for load volumes you may anticipate in the future.

Tuning methodology

There no single database tuning methodology is optimal or complete for every Oracle Server configuration. One of the common tuning approaches to accomplish the greatest increase in performance with the least amount of effort consists of the following four phases:

1. Checking/tuning the initialization parameters (INIT.ORA). You can use [DB Tuning Expert](#) for this purpose.
2. Running and recording benchmarks. You can use [DB Benchmark Expert](#) for this purpose.
3. Monitoring and archiving performance data. You can use [DB Monitor Expert](#) and [DB Trends](#) for this purpose.
4. Optimizing end user SQL and schema objects. You can use such powerful tools (from [DB Monitor Expert](#) package) as [DBA Notepad](#), [Explain Plan](#), [SQL Area monitor](#), [Connections monitor](#), [Connection Manager](#), and [Dead Lock Analyzer](#) for this purpose. You can also use [DB Index Expert](#) and [DB Tuning Expert](#) for this purpose.

Performance Analysis Options

Refer to the following tables for analysis options:

Options	Description
Analyze instance parameters and database configuration	<ul style="list-style-type: none"> Analyze statistics from Oracle dynamic performance tables Analyze database logical and physical models including number of files, tablespaces, rollback segments, tablespace fragmentation, and many other
Analyze schema objects	<ul style="list-style-type: none"> Analyze object space allocation and storage parameters Analyze object fragmentation Analyze object logical design
Analyze offensive SQL from the Library Cache	<ul style="list-style-type: none"> Search the Library Cache for offensive SQL queries (SQL causing most buffer gets and physical I/Os) Analyze found offensive queries for better execution plans
Analyze database trends (data by DB Trends is required)	<ul style="list-style-type: none"> Analyze historical trend data collected by DB Trends.

Report Output Formats

The resulting performance analysis report can be generated in one of the following formats:

- Print Preview
- Enhanced hypertext mark-up language (HTML) file format. You need a Web browser that supports frames in order to view and print reports. We recommend Netscape Navigator 3.x and above or

Microsoft Internet Explorer 3.x and above. After report generation, Tuning Expert automatically launches your default Web browser.


- ASCII text-format file that can be read in Windows Notepad and any other word-processors. The report is created in the DB Tuning Expert installation directory.

DB Space Expert

Overview

DB Space Expert can be used to reorganize, optimize, partition and consolidate objects within databases while the database applications remain up and running and fully available to users. DB Space Expert can be used to quickly identify objects that would benefit most from space reorganization and then reorganize them either in a real-time or schedule reorganization jobs for those objects or even entire tablespaces. By optimizing object storages, defragging tablespaces and files, and reclaiming wasted disk space, DB Space Expert reduces the need for additional storage requirements and helps to avoid expensive storage upgrades.

It does not matter how small or big your database is, how complex your environment is or how many user you have. DB Space Expert can always help you keep your database running at peak performance.

 **Important Note:** DB Space Expert can be used with any Oracle database version 7.3 and up. However some functions, such as partitioning, advanced scheduling, and space recycling are not available in old database versions.

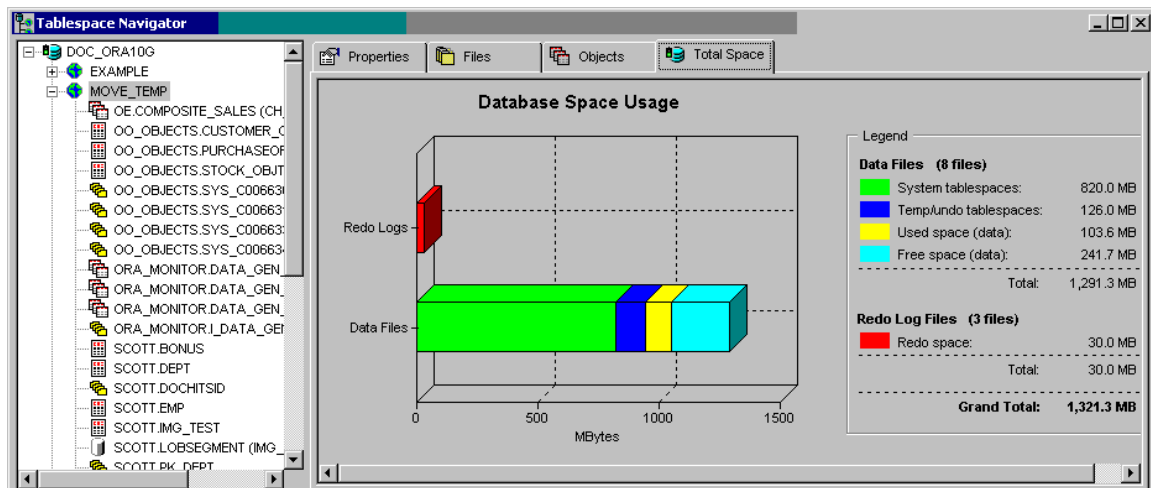
Using DB Space Expert's Interface

Tablespace Navigator

The Tablespace Navigator appears as both a tablespace browser tool and as an entry point to most other tools. The Table Navigator displays your database organization from a space usage point of view and allows drill-down from tablespaces to storage objects and their properties.

The Tablespace Navigator window is divided into two parts:

- The "Navigator" - left pane
- The "Object Properties" - right pane



You can change the size of the panes. When you make any pane larger, the adjacent pane becomes smaller. To resize panes:

1. Move the mouse pointer in between of the panes until you see a black line with two arrows.



2. Drag the edge of the pane until the pane is the size you want.

Right-clicking on an item in the Navigator pane brings up the context popup menu with most common functions available for the type of the clicked object, for example, move object to another tablespace, split partition, deallocate unused space and so on...

Right-clicking on the properties page, graph or report displayed in Object Properties pane brings up the context popup menu with most common functions available for the clicked document type, for example, print, zoom, export, filter and so on...

Starting Tablespace Navigator

The DB Space Expert automatically opens the Tablespace Navigator after a database connection is established. If you managed to close the Tablespace Navigator window use Space/Tablespace Navigator menu (shortcut F7) to reopen it.

Navigator Views and Properties

Database

Database item is always the top-level item in the navigator tree-view. The following views are available when the database item is selected:

- Tablespaces – displays free/used space allocated chart for by tablespace and summary allocation table. Tablespaces having less than 10 % of free space highlighted in red.
- Properties – displays general database properties and version information
- Files – displays list all database files and their properties, including name, status, size, auto-extensible status, increment size and tablespace name
- Total Space – displays overall database space usage chart grouped by different types of files and free space

Tablespace

Tablespaces are displayed on the second level of the Navigator tree-view as "children" of the top-level database item. The following views are available when a tablespace item is selected:

- Properties – displays tablespace properties. This page different for different Oracle versions. More recent versions have more properties than older versions. See your Oracle documentation for detailed description of tablespace properties.
- Files – displays list all files the selected tablespace consists of and their properties, including name, status, size, auto-extensible status, increment size and tablespace name
- Objects – displays all databases objects stored in the selected tablespace, including their types, names, extents, sizes and parent objects.
- Total Space – displays overall database space usage chart grouped by different types of files and free space

Schema Object

Schema objects are displayed on the third level of the Navigator tree-view as "children" of the selected tablespace item. Different icons are used to represent different types of schema object such as tables, overflow tables, IOT tables, nested tables, IOT mapping tables, snapshots, snapshot logs, materialized views, materialized view logs, table partitions, indexes, index partitions, LOB indexes, LOB segments (LOB columns, Java classes, Java resources, other), LOB partitions, clusters.

The following views are available when a schema object-item is selected:

- Properties – displays object properties. Properties differ for different object types and Oracle versions.
- Extents – displays information about all extents the selected object consists of, including file names, extent sizes, and extent IDs
- Objects – displays all databases objects stored in the selected tablespace, including their types, names, extents, sizes and parent objects.
- Total Space – displays overall database space usage chart grouped by different types of files and free space

Context Menus

DB Space Expert's top-level Space and Partitions menus are sensitive to the type of an item currently selected in the Tablespace Navigator. Depending on the selected item type such DB Space Expert enables or disables different commands in these menus.

Similarly, the Navigator's right-click popup menu reacts differently to different item types.

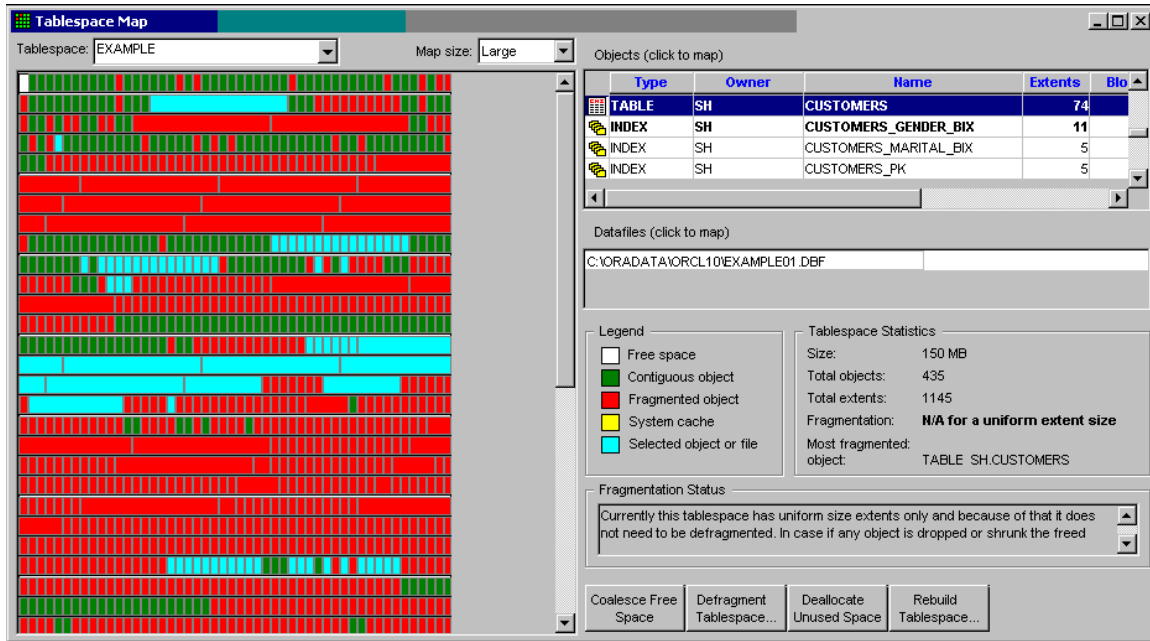
Try clicking on different items to see how it works.

Working with Tablespaces and Files

Tablespace Maps

Overview

You need to continually inform yourself of tables in your database that are overextended or are reaching their maximum extents limit or have high degree of fragmentation. The Tablespace Map utility helps to bring this information to your attention now before extent problem occurs and helps to keep your database tuned. It lets you view a graphical layout regarding the physical placement of every object inside a selected tablespace as well as pockets of free unallocated space.



The Tablespace Map utility also helps you identify fragmentation problems for the entire tablespace.


In the **Objects and Files** section the Tablespace Map utility displays the complete interactive list of all tablespace segments and files with detailed information about every file and object's storage properties.

The **Tablespace Statistic** section presents summary tablespace information including tablespaces fragmentation status, total size, most fragmented object and so on.

A separate **Fragmentation Status** box presents additional comments about tablespace fragmentation status and whether any action is recommended at this time.

A sample screenshot of the Tablespace Map is present above.


Using the Tablespace Map utility

1. Start Tablespace Map utility from **Reports/Tablespace Map** menu (shortcut CTRL N) or simply click the Map  button on the application toolbar. The tablespace map window will open.
2. Choose the required tablespace from the **Tablespace** drop-down list box.
3. When painting of space allocation for the tablespace is complete, you can pick an object in the object list or pick a tablespace file in the file list for which you want to know its extents allocation within the tablespace. The selected object or file will become highlighted on the map.
4. To change the level of details presented on the tablespace map use **Map size** dropdown. If you have large tablespaces with relatively small number of objects pick Small size; otherwise pick Large or Medium size.



Tip: For your convenience the Tablespace Map utility provides several buttons with one click access to commonly used tablespace reorganization functions and wizards.

Creating New Tablespaces

Click **Space/Create New Tablespace** menu to open **New Tablespace** dialog. Alternatively, you can click the Create  button on the application toolbar.

New Tablespace

General (7.3+) Oracle 8.0 Oracle 8i Oracle 9i Oracle 10g

Enter tablespace name, content type and status:
 Name: Content Type: Status:


Specify one or more tablespace files:

File Name (including path)	File Size (MB)	Auto- extensible?	Max File Size (MB)	Increment By (MB)	Reuse?
C:\ORADATA\ORCL10\NEW_TS.DBF		<input type="text" value="YES"/>	<input type="text" value="UNLIMITED"/>	50	<input type="text" value="NO"/>
	0		UNLIMITED		

Specify default storage parameters for new tablespace objects (tablespaces with non-TEMPORARY content only):

INITIAL (KB):
 NEXT (KB):
 MINEXTENTS:
 MAXEXTENTS: Allow UNLIMITED extents
 PCTINCREASE (%):

Note: Use 0 value for PCTINCREASE to set all default segments to the same size and minimize tablespace fragmentation.

 Some options appear disabled in case if they are incompatible with other selected options. For example, In Oracle 8i and later default storage options cannot be used when tablespace extent management type is set to LOCAL. Changing this type to DICTIONARY will allow you to use storage options and make them enabled.

The **New Tablespace** dialog consists of several tab pages. Different tab page are used with different versions of Oracle databases.

The first page labeled **General (7.3+)** is a common page available in all databases. If your database version is after 7.3 then one more page becomes enabled in addition to the **General** page. This page contains your database version specific properties. The following example presents the Oracle 10g page.

New Tablespace

General (7.3 +) Oracle 8.0 Oracle 9i Oracle 10g

Specify default logging options for new tablespace objects (tablespaces with PERMANENT content only):

Logging: YES Force logging: NO

Specify tablespace space usage parameters:

Minimum extent size (KB): 40 Block size (KB): 8 Table Compression: NO

Specify tablespace extent management type:

Type: LOCAL AUTOALLOCATE Uniform extent size (KB): 80

Specify tablespace segment management type (tablespaces with PERMANENT content, LOCAL extent management only):

Type: AUTO

Specify tablespace flashback mode (tablespaces with PERMANENT content only):

Flashback mode: ON

Specify tablespace group membership (tablespaces with TEMPORARY content only):

Group name: Note: leave this parameter blank to indicate that tablespace is not a member of any tablespace group.

Specify tablespace retention type (tablespaces with UNDO content only):

Retention type: NOGUARANTEE

Create Cancel

By default all properties are preset to Oracle defaults. You must fill in at least 2 properties on the **General** page in order to create a new tablespace, which are tablespace name and file size.


Note: As you type new tablespace name DB Space Expert atomically suggests new file name for the tablespace so that you don't have to type it. However you can modify the suggested file name as needed and also add additional files using the Insert Row button. To delete mistakenly added file use the Delete Row button.

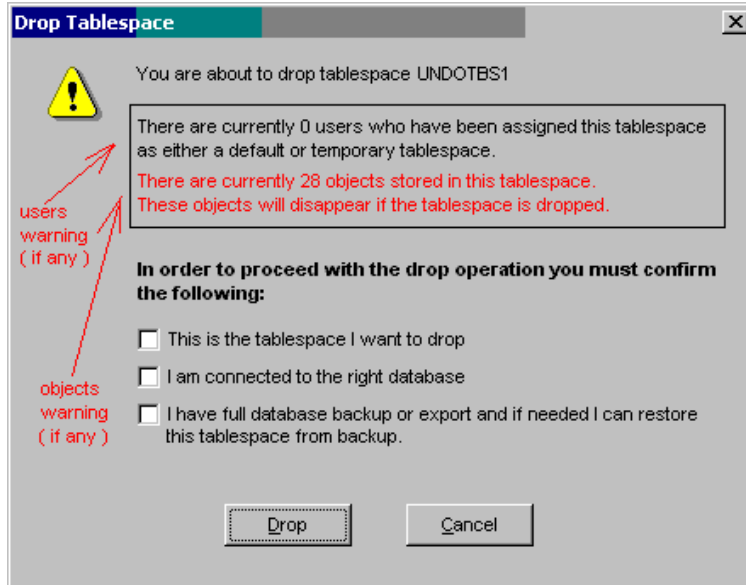
Tip: See Oracle documentation for detailed description of tablespace properties and usage modes.


Taking Tablespaces Off-line/On-line

1. In the [Tablespace Navigator](#) select the tablespace whose off-line/on-line state you want to toggle.
2. Click the **Space** menu and then click the **Take Off-line/Place On-line** command to toggle status of the selected tablespace. Alternatively, you can click the Offline button on the application toolbar. If you are taking tablespace off-line a message prompt will appear asking to confirm this action. If you are sure click the YES button to take selected tablespace off-line.

Dropping Tablespaces


1. In the [Tablespace Navigator](#) select the tablespace you want to drop.
2. Click the **Space** menu and then click the **Drop Tablespace** command. Alternatively, you can click the Drop  button on the application toolbar. This will display the **Drop Tablespace** dialog.



 **Tip:** DB Space Expert automatically checks tablespace dependencies and if warns you if the tablespace is still used. It displays in red any critical warnings. You can choose to ignore them if you are sure or you can cancel the **Drop Tablespace** dialog and perform corrective actions first.


3. For your own protection you must check all free checkboxes displayed on the dialog:
 - **This is the tablespace I want to drop**
 - **I am connected to the right database**
 - **I have full tablespace backup or export and if needed I can restore this tablespace from a backup.**
4. Click the **Drop** button to drop the selected tablespace.

Adding Files to Tablespaces


1. In the [Tablespace Navigator](#) select the tablespace you want to add files to.
2. Click the **Space** menu and then click the **Add File To Tablespace** command. Alternatively, you can click the Add  button on the application toolbar. This will display the **Add New File** dialog.
3. Enter new file name and size parameters. If you don't want to use the auto-extend option, uncheck the **Autoextensible** option
4. Click the **Add File** button to create a new file, pre-allocate specified file space and add this new file to the selected tablespace.

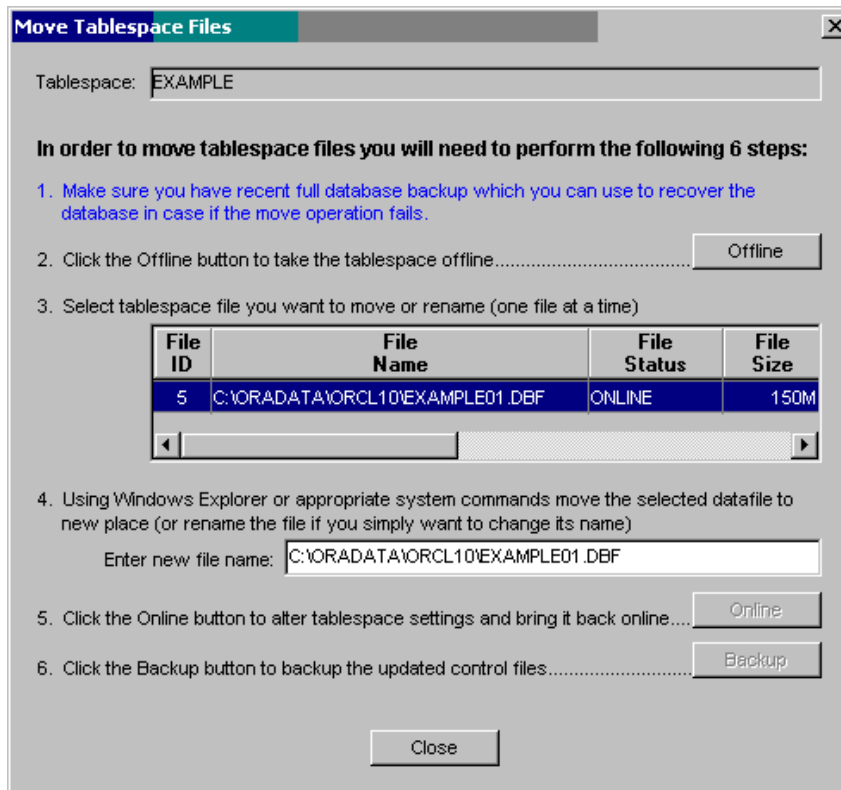
Resizing Tablespaces

1. In the [Tablespace Navigator](#) select the tablespace whose files you want to resize.
2. Click the **Space** menu and then click the **Resize Tablespace** command. Alternatively, you can click the **Resize**  button on the application toolbar. This will display the **Resize Tablespace File** dialog.
3. If the tablespace consists of multiple files, select the file you want to resize
4. Enter new file size parameters. In case if you want Oracle to automatically expand this file when it runs out of space check the **Autoextensible** option and specify increment size for new file chunks.
5. Click the **Resize File** button to modify file properties.

 **Note:** You can both increase and decrease file sizes. When decreasing sizes make sure that the new file size is big enough to fit all current objects stored in that file and the size change delta value is not bigger than the size of the ending free space extent. You can use the [Tablespace Map](#) utility to figure out how much free space is available database tablespaces and files.

Moving and Renaming Database Files

1. First of all make sure you have a full database backup that you can use to restore the tablespace if the file move or rename operation fails for whatever reason.
2. In the [Tablespace Navigator](#) select the tablespace whose files you want to move or rename.
3. Click the **Space** menu and then click the **Move/Rename Tablespace File(s)** command. Alternatively, you can click the **Move**  button on the application toolbar. This will display the **Move Tablespace Files** dialog.



Move Tablespace Files

Tablespace:

In order to move tablespace files you will need to perform the following 6 steps:

1. Make sure you have recent full database backup which you can use to recover the database in case if the move operation fails.
2. Click the Offline button to take the tablespace offline.....
3. Select tablespace file you want to move or rename (one file at a time)


File ID	File Name	File Status	File Size
5	C:\ORADATA\ORCL10\EXAMPLE01.DBF	ONLINE	150M

4. Using Windows Explorer or appropriate system commands move the selected datafile to new place (or rename the file if you simply want to change its name)

Enter new file name:

5. Click the Online button to alter tablespace settings and bring it back online....

6. Click the Backup button to backup the updated control files.....

4. Click the **Offline** button to take the tablespace offline. Note that if this operation is successful the **Offline** button becomes disabled and the **Online** button becomes enabled.
5. If the tablespace consists of multiple files, select the file you want to move or rename
6. Using any appropriate operation system commands to physically rename or move the selected file to a different location. **Do not rename/move multiple files at once! You must rename/move one file at a time.** In case if you want to rename or move multiple files, complete the following steps 7, 8 and 9 and then repeat steps from 4 to 9 for another file.
7. Enter new file name in to the **New File Name** edit field.
8. Click the **Online** button to bring the tablespace online.
9. Click the **Backup** button to backup the updated control files..
 **Note:** The backup button does NOT backup your database files and it does not write your existing database backups to a tape or some other place. It simply creates updated copies of the database control files that Oracle can use later to startup the database in case if the main control files become corrupted.
10. Repeat steps 4 to 9 for other files in the same tables otherwise click the **Close** button to close this dialog.

 **Important Note:** The **Move Tablespace Files** dialog cannot be used to move or rename files for the SYSTEM tablespace.

Converting Tablespaces

Starting with version 8.1.5 Oracle supports "locally managed tablespace" that can manage extent allocation within the tablespace without using data dictionary tables. These tablespaces maintain a bitmap in each datafile to keep track of the freed or used status of blocks in that datafile. Each bit in the bitmap corresponds to a block or a group of blocks. When an extent is allocated or freed for reuse, Oracle changes the bitmap values to show the new status of the blocks. As opposite to the data dictionary managed tablespaces these changes do not generate rollback information because they do not update tables in the data dictionary (except for special cases such as tablespace quota information). Oracle supports both types of tablespaces, however Oracle recommends using locally managed tablespaces because they are more efficient.

If you have upgraded your database from version 8.0 or prior you may want to convert existing data dictionary tablespaces to locally managed tablespaces. DB Space Expert provides an automated tool for that purpose. To perform the conversion do the following:

1. In the Tablespace Navigator select data dictionary managed tablespace which you want to convert to locally managed tablespace.
2. Click the **Space** menu and then click the **Convert To Locally Managed** command. This will display a prompt to confirm the operation.
3. Click **OK** button if you are sure. DB Space Expert will convert the selected tablespace.


Important Notes:

- The tablespace being converted cannot be used during conversion processing. This applies to all objects contained in the tablespace.
- You cannot use this method to convert the SYSTEM tablespace.
- You cannot also use it for tablespaces that contain active rollback segments.
- You cannot also use it for tablespaces having TEMPORARY tablespace type.


Modifying Default Storage Parameters

To modify default storage parameters for new objects that will be created in the selected tablespace without specific storage clauses or partial storage clauses do the following:

1. In the [Tablespace Navigator](#) select the tablespace whose default storage parameters you want to modify.
2. Click the **Space** menu and then click the **Edit Default Storage Parameters** command. This will display the **Edit Tablespace Storage Parameters** dialog.
3. Modify parameters as required. The **Edit Tablespace Storage Parameters** dialog displays detailed descriptions of all parameters and also helpful tips and notes.
4. Click the **OK** button to modify tablespace properties.

 **Note:** Changing tablespace default storage parameters does NOT affect existing objects nor it affects new objects in a tablespaces having uniform space allocation type. It only affects how space is managed for new objects (created after this change) in tablespaces with non-uniform extent allocation types.

Managing User Space Quotas and Defaults

To view or modify user space quotas click **Space** then select **Edit Space Usage Quotas** menu or click the Space Quota  button on the application toolbar. The **Tablespace Quotas** dialog will appear.

This dialog consists of two tab pages each providing a different view to the current database configuration:


- The first tab page **Space Quotas by User** displays space quotas grouped by user, in other words, for every user it shows all granted tablespaces and space quotas.
- The second tab page **Space Quotas by Tablespace** displays space quotas grouped by tablespace name, in other words, for every tablespace it shows all users who has been granted space in the tablespace space and their quotas.

You can pick and modify any view you want. Use whatever is more convenient to you. The end result will be the same.


To modify an existing user space quota:

1. Activate **Space Quotas by User** tab page.
2. Click on the user whose tablespace quotas you want to modify in the left hand side tree-view
3. On the right hand side of the screen in the table-like grid control modify existing quotas.

or

To add another tablespace click the Insert  button on any existing line. A new row will appear above the clicked button. In the Tablespace Name column select required tablespace name in the drop-down list. Enter the required quota in the Space Quota column. The quota must be either a valid number or one of the following values: [DEFAULT], [UNLIMITED].

or


To completely revoke an existing tablespace quota use the Delete  button on the tablespace line whose quota you want to revoke from the user.

The top screenshot shows the 'Tablespace Quotas' dialog box with the 'Space Quotas By User' tab selected. The tree view on the left shows the following users: BI, SYS, TEMP, and USERS. The table on the right displays the following data:

Tablespace Name	Space Type	Quota (MBytes)	Used (MBytes)
SYS	NORMAL	100	
TEMP	TEMPORARY	[UNLIMITED]	
USERS	DEFAULT	[UNLIMITED]	
Total:		[UNLIMITED]	0

The bottom screenshot shows the 'Tablespace Quotas' dialog box with the 'Space Quotas By Tablespace' tab selected. The tree view on the left shows the following users: EXAMPLE, MOVE_TEMP, and SYS. The table on the right displays the following data:


User Name	Space Type	Quota (MBytes)	Used (MBytes)
ANONYMOUS	DEFAULT	[UNLIMITED]	
BI	NORMAL	100	
CTXSYS	DEFAULT	[UNLIMITED]	5
DBSNMP	DEFAULT	[UNLIMITED]	0
DMSYS	DEFAULT	200	5
EXFSYS	DEFAULT	[UNLIMITED]	1
MDSYS	DEFAULT	[UNLIMITED]	6
MGMT_VIEW	DEFAULT	[UNLIMITED]	
OLAPSYS	DEFAULT	[UNLIMITED]	15
ORDPLUGINS	DEFAULT	[UNLIMITED]	
ORDSYS	DEFAULT	[UNLIMITED]	1
SI_INFORMTN_SCHEMA	DEFAULT	[UNLIMITED]	
SYSMAN	DEFAULT	[UNLIMITED]	48
WKPROXY	DEFAULT	[UNLIMITED]	
WKSYS	DEFAULT	[UNLIMITED]	7
WK_TEST	DEFAULT	[UNLIMITED]	12
WMSYS	DEFAULT	[UNLIMITED]	7
XDB	DEFAULT	[UNLIMITED]	47
Total:		[UNLIMITED]	154


 **Tip:** The UNLIMITED quota means to Oracle that the user can allocate as much space as much free space is available in the tablespace. If the tablespace files have AUTOEXTENT option turned on and the user requests more space than available Oracle automatically increases size of tablespace files to accommodate the new space requirements. The DEFAULT quota means to Oracle that the database server can decide how much space it can make available to the user.

- Click the **Apply** button to save all changes.

Coalescing Tablespace Free Space

If you have free space fragmentation, you can use the Coalesce Free Space utility to consolidate free space extents into larger chunks of free space.

 **Background Information:** Free space fragmentation is a situation that develops in a tablespace, where "holes" appear in the tablespace. These holes are made up of free blocks, but no objects attempt to use them when they need to extend. They are created as a result of adding and deleting extents from the objects in the tablespace, where these objects have different initial and/or next extent sizes. Having the object or the tablespace use a non-zero PCTINCREASE value is also a great way to cause this problem as well. There is no known need to have the PCTINCREASE setting anything other than zero if you are running Oracle version 8.0 or later and have set maximum extent number for all objects to UNLIMITED value. However, if you haven't done that and experience free space fragmentation, do the following to coalesce adjacent free space extents:

1. In the [Tablespace Navigator](#) select the tablespace you want to coalesce.
2. Click the **Space** menu and then click the **Coalesce Tablespace's Free Space** command. Alternatively, you can click the Coalesce  button on the application toolbar. DB Space Expert will coalesce free space and display a status message when done.

**Tips:**

- Coalesce Free Space utility is only able to coalesce adjacent free extents. This operation does not affect any existing objects and is very fast
- To coalesce non-adjacent free extents (e.g. "holes" of free space appearing in the middle of the tablespace) use the [Reorganize / Defrag Tablespace Wizard](#) utility.
- The [Tablespace Map](#) utility provides graphical display of tablespace space allocation that shows free space fragmentation in an convenient easy-to-understand format

Estimating Total Unused Space in All Objects



Background Information: Oracle automatically allocates new extents for an object when the objects requires more storage space. However, Oracle does not automatically release (deallocate) extents when an objects space requirements fall as a result of data deletion, index reorganization and so on. As a result a significant amount of unused space can be developed.

DB Space Expert provides a method to [deallocate unused space](#) in all tablespace objects in a single operation, in other words to reclaim tablespace unused.

In case if you want to estimate how much free space can be deallocated without actually running the [Free Unused Space](#) utility do the following:

1. In the [Tablespace Navigator](#) select the tablespace you want to analyze.
2. Click the **Space** menu and then click the **Estimate Total Unused Space in All Objects** command. DB Space Expert will check all tablespace objects, estimate unused space in each object and then display a status message with the estimated summary results.


Freeing Unused Space in All Objects



Background Information: Oracle automatically allocates new extents for an object when the object requires more storage space. However, Oracle does not automatically release (deallocate) extents when object's space requirements fall as a result of data deletion, index reorganization and so on. As a result a significant amount of unused space can be developed.

DB Space Expert provides a method to [deallocate unused space](#) in all tablespace objects in a single operation, in other words to reclaim tablespace unused space.

In case if you want to estimate how much free space can be deallocated without actually running the [Free Unused Space](#) utility do the following:

1. In the [Tablespace Navigator](#) select the tablespace you want to reorganize.
2. Click the **Space** menu and then click the **Free Unused Space in All Objects** command. Alternatively, you can click the Free Unused  button on the application toolbar. DB Space Expert will check all tablespace objects and then free unused space in all objects having significant amount of unused space. On completion DB Space Expert will display a status message with the summary results describing total amount of tablespace space it was able to reclaim.

**Tips:**

- In case if want to estimate how much space can be reclaimed and whether it is worth to processed with the Free Unused Space option, use the [Estimate Total Unused Space in All Objects](#) option.
- In tablespaces with uniform extent size allocation methods you can have a situation when unused space cannot be released because of the freed space having non-uniform extent size. Even though the "Estimate" will report a significant amount of unused space you will not be able to reclaim it.
- You can use the [Reorganize / Defrag Tablespace Wizard](#) utility to coalesce free space extents released as a result of this processing.

Emptying Tablespace Recycle Bin

By default Oracle 10g databases and up use the flush-back feature to register dropped and modified objects in the database Recycle Bin area without actually freeing the allocated disk space so that such objects can be restored later. To free the Recycle Bin for a particular tablespace and reclaim the free space do the following:

1. In the [Tablespace Navigator](#) select the tablespace whose objects you want to permanently remove from the database recycle bin.
2. Click the **Space** menu and then click the **Empty Tablespace Recycle Bin** command. DB Space Expert will permanently delete all recycled objects and free their space for other objects in the selected tablespace.




Tips:

- You can use the [Reorganize / Defrag Tablespace Wizard](#) utility to coalesce free space extents released as a result of this processing.
- You can use the [Explore Recycle Bin](#) utility to browse contents of the database Recycle Bin and purge individual objects or groups of objects.

Reorganizing and Defragging Tablespaces


To reorganize space allocation on-line, defrag segments and free space extents in a particular tablespace do the following:

1. In the [Tablespace Navigator](#) select the tablespace you want to reorganize.
2. Click **Tools** menu and then click **Reorganize / Defrag Tablespace Wizard** command. Alternatively you can click the Reorganize  button on the application toolbar. This will start the **Reorganize / Defrag Tablespace Wizard** utility and automatically enter the selected tablespace as the operation target. Follow instructions provided by the Wizard to complete the operation. For more information and usage instructions read [Reorganize / Defrag Tablespace Wizard](#) topic.

Rebuilding Tablespaces


To rebuild an existing tablespace on-line using different storage parameters, files or location do the following:

1. In the [Tablespace Navigator](#) select the tablespace you want to rebuild.

2. Click **Tools** menu and then click **Rebuild Tablespace Wizard** command. Alternatively you can click the Rebuild  button on the application toolbar. This will start the **Rebuild Tablespace Wizard** utility and automatically enter the selected tablespace as the operation target. Follow instructions provided by the Wizard to complete the operation. For more information and usage instructions read [Rebuild Tablespace Wizard](#) topic.

Moving Tablespace Objects to Another Tablespace

DB Space Expert supports easy graphical methods for moving objects between tablespaces.


1. In the [Tablespace Navigator](#) select the object you want to move.
2. Click **Tools** menu and then click **Move Objects Wizard** command. Alternatively you can click the Move Object  button on the application toolbar. This will start the **Move Objects Wizard** utility and automatically enter the selected objects as the operation target. Follow instructions provided by the Wizard to complete the operation. For more information and usage instructions read [Move Objects Wizard](#) topic.

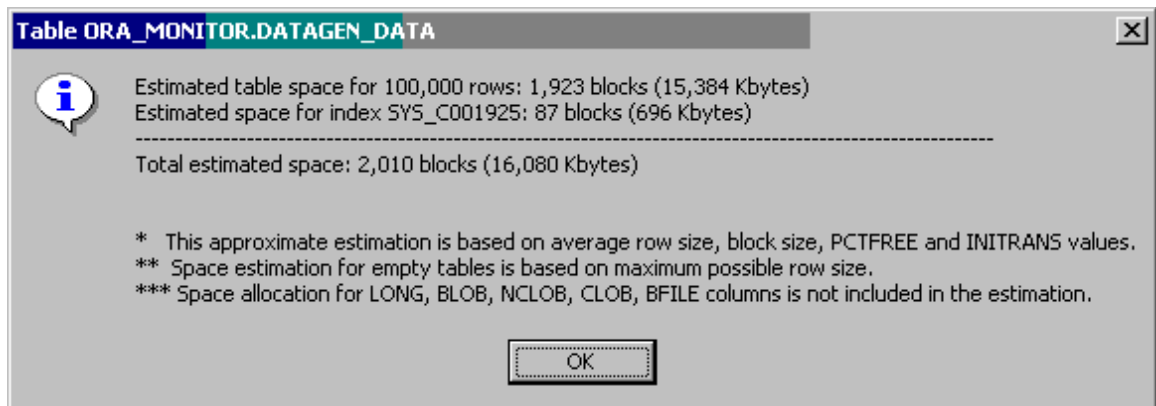
Working with Schema Objects

Estimating Table and Index Space Requirements

Use the **Table Size Estimator** utility to project how much space will be required for storing certain number of records on a specific table, partition or index.

To access this utility use

1. In the [Tablespace Navigator](#) select the object whose space you want to project.
2. Click the **Space** menu and then click the **Table Size Estimator** command. DB Space Expert will display the **Table Size Estimator** dialog. Alternatively you can click the Estimate  button on the application toolbar. Note that in case if an index or partition is selected in the Tablespace Navigator, the Table Size Estimator automatically finds the parent table and displays this table name on the dialog.
3. Enter projected row count in the **Number of Rows to Estimate** field.
4. Click the **OK** button. DB Space Expert will perform table analysis and space estimation. After that, a message box will appear where you will see space estimation results for the table data and each index separately as well as the total amount of estimated space.




Dropping Schema Objects

To drop an object from the database:

1. In the [Tablespace Navigator](#) select the object you want to drop.
2. Click the **Space** menu and then click the **Drop Object** command. DB Space Expert will ask you to confirm that the right object is selected for deletion and if confirmed it will drop that object.

Moving Schema Objects Between Tablespaces

DB Space Expert supports easy graphical methods for moving objects between tablespaces.


1. In the [Tablespace Navigator](#) select the object you want to move.
2. Click **Tools** menu and then click **Move Objects Wizard** command. Alternatively you can click the Move Object  button on the application toolbar. This will start the **Move Objects Wizard** utility and automatically enter the selected objects as the operation target. Follow instructions provided by the Wizard to complete the operation. For more information and usage instructions read [Move Objects Wizard](#) topic.

Rebuilding Indexes

To rebuild all indexes index for a particular table or table partition:

1. In the [Tablespace Navigator](#) select the object whose indexes you want to rebuild.
2. Click the **Space** menu and then click the **Rebuild Indexes** command. DB Space Expert will find and rebuild all object indexes using Oracle's standard ALETER INDEX...REBUILD command.

Freeing Object's Unused Space

 **Background Information:** Oracle automatically allocates new extents for an object when the object requires more storage space. However, Oracle does not automatically release (deallocate) extents when object's space requirements fall as a result of data deletion, index reorganization and so on. As a result a significant amount of unused space can be developed.

DB Space Expert provides a method to deallocate unused space and free for use in other objects.


1. In the [Tablespace Navigator](#) select the object you want to reorganize.
2. Click the **Space** menu and then click the **Free Unused Space** command. DB Space Expert will free unused space in the selected object. On completion DB Space Expert will display a status message with the summary results describing total amount of space it was able to reclaim.

Tips:

- In tablespaces with uniform extent size allocation methods you can have a situation when unused space cannot be released because of the freed space having non-uniform extent size.
- You can use the [Free Unused Space in All Objects](#) utility to free unused space in all tablespace objects in a single operation.

Fixing Chained Rows


Row Chaining and Migration are common problems in Oracle database tables. Row migration is caused by rows whose data grows in length over time and at some point it can no longer fit into its original Oracle block (block allocated for the original row data). As a result Oracle has to allocate additional blocks to fit the rest of the row and during I/O operations read/write more data blocks. Row chaining is not about the growth of rows data, but is the result of rows data starting out too big.

If your tables developed such problems you can fix them using **Fix Chained Rows Wizard**. Use **Tools/Fix Chained Rows Wizard** menu to start this Wizard. Alternatively you can click the Fix Chains  button on the application toolbar.

See [Fix Chained Rows Wizard](#) topic for more information and usage instructions.


Working with Partitions


DB Space Expert provides a complete set of graphical tools for managing table partitions. These tools can be used to partition and unpartition tables, split and merge partitions, rename and move them, rebuild and exchange data with standalone tables.

 **Note:** Because of the Oracle 7 lack of support for partitions, DB Space Experts partitioning tools are only enabled when connected to Oracle 8.0 databases and later.

The following topics describe in detail how to use these tools.

Converting Regular Tables to Tables with Partitions

 **Note:** Tables with LONG and LONG RAW columns, index-organized tables and nested tables cannot be partitioned.

1. In the [Tablespace Navigator](#) select an existing table you want to reorganize and convert to partitioned table.
2. Click the **Partitions** menu and then click the **Partition Table** command. Alternatively you can click the Partition  button on the application toolbar. This will open the **Convert Regular Table To Partitioned Table** dialog. An example screenshot of the dialog follows.

Convert Regular Table to Partitioned Table

Current Table


Owner: Table:
 Partition: Tablespace:

Partitioning Type: By Range By Hash By List

Number of Partitions: Partition Key Column(s):

Partitions:

	Less Than Value	Partition Name	Tablespace
1.			TOOLS
2.	MAXVALUE		TOOLS

 String and date values must be entered in single quotes, for example '01-JAN-2004'. You can also enter expressions like TO_DATE('2004-01-01', 'YYYY-MM-DD', 'NLS_CALENDAR=GREGORIAN')

Processing Options

Use PARALLEL option for faster processing
(tables with object columns or LOB columns don't support this option; this option may also cause overallocated space for "small" objects)


Use UNRECOVERABLE and NOLOGGING options for faster processing

Rebuild global and local indexes after table is converted

Rebuild object statistics after the object is modified
(run *smart* ANALYZE TABLE/INDEX commands after table is converted; this option increases processing time but ensures the database can use best query optimization plans)

Recompile dependent code objects
(recompile dependent PROCEDURES, TRIGGERS, PACKAGES, etc..., after table is converted)


- Select **Partition Type** option. Note that partition **By Range** option is available in all Oracle databases (8.0 and up); **By Hash** is available in Oracle 8i and up; **By List** is enabled in Oracle 9i and up.
- Enter **Number of Partitions** value. You can type in the required number or you can use arrows up and down to increment/decrement the value as much as needed.
- Enter **Partition Key Column(s)**, in other words names of table columns whose values define where partitions start and end.


 **Note:** If you are planning to use a single column for the key you can select its name from the drop-down list, otherwise you must enter column names as a comma-separated list. In the latest case make sure you enter valid column names in the proper order. If you select **By List** partitioning type you must use single column as a partition key
- Enter partition names, storage parameters and partition range/list specification into **Partitions** grid. Note that the dialog shows different partition properties for different partition types.

If **By Range** partition type is chosen then ranges are used for partitioning table data. Ranges must be defined for each partition using the values less than (value) clause, and the value specified in the cannot include a volatile data operation, such as calls to non-standard functions. String and date values must be enclosed in single quotes, as you would use them in other SQL expressions.

	Less Than Value	Partition Name	Tablespace
1.	'31-DEC-2002'	P_LESS_31_DEC_2002	USERS
2.	'31-DEC-2003'	P_LESS_31_DEC_2003	USERS
3.	'31-DEC-2004'	P_LESS_31_DEC_2004	EXAMPLE

- 4.1. Enter partition names and expressions for **less than (value)** clause. Partition names must conform to Oracle object naming standards. To avoid partitioning problems the last partition value is automatically entered for you as MAXVALUE keyword, which means "catch-all" partition for all key values that are not in ranges entered for other partitions.
- 4.2. Select partition locations using the **Tablespace** drop-down list.
- 4.3. Enter partition storage parameters in case if you do not want to use tablespace defaults.


To edit storage parameters click the Storage  button. You can select different tablespaces and storage parameters for different partitions.


 **Note:** Entered storage parameters will be ignored if the selected tablespace is a locally managed tablespace with automatic extent management.

If **By Hash** partition type is chosen then Oracle automatically calculates data ranges for data partitioning and you only enter partition names, locations and storage parameters.

	Partition Name	Tablespace
1.	P_1	USERS
2.	P_2	USERS
3.	P_3	EXAMPLE

- 6.1. Enter partition names. Partition names must conform to Oracle object naming standards.
- 6.2. Select partition locations using the **Tablespace** drop-down list
- 6.3. Enter partition storage parameters in case if you do not want to use tablespace defaults.


To edit storage parameters click the Storage  button. You can select different tablespaces and storage parameters for different partitions.

 **Note:** Entered storage parameters will be ignored if the selected tablespace is a locally managed tablespace with automatic extent management.


If **By List** partition type is chosen then you must enter list of key values for each selected partition.

	Values	Partition Name	Tablespace
1.	'AK', 'AL', 'AR', 'AZ'	P_A_STATES	USERS
2.	'CA', 'CO', 'CT'	P_C_STATES	USERS
3.	'DE', 'FL'	P_LESS_DE_FL	EXAMPLE


- 6.1. Enter partition names and list values. Partition names must conform to Oracle object naming standards. To avoid partitioning problems the last partition value is automatically entered for you as DEFAULT keyword, which means "catch-all" partition for all key values that are not listed in other partitions.
- 6.2. Select partition locations using the **Tablespace** drop-down list
- 6.3. Enter partition storage parameters in case if you do not want to use tablespace defaults.

To edit storage parameters click the Storage  button. You can select different


tablespaces and storage parameters for different partitions.

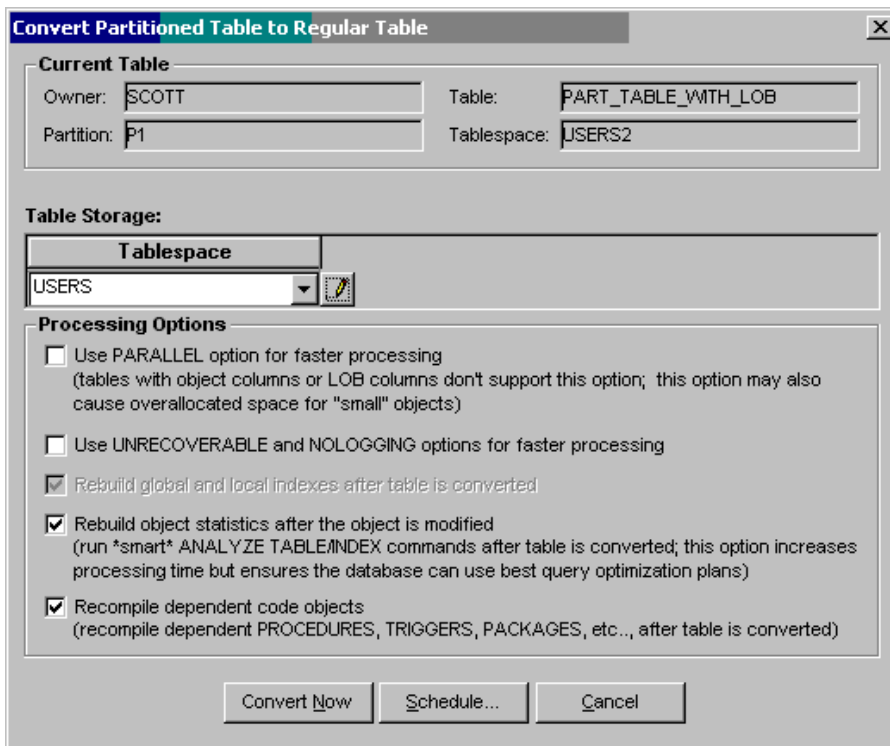
 **Note:** Entered storage parameters will be ignored if the selected tablespace is a locally managed tablespace with automatic extent management.

7. In the **Processing Options** box select advanced processing options for the conversion operation
8. To start the conversion operation immediately on the front-end script, click the **Convert Now** button. To schedule this operation for a later run, perhaps off-hours, or to start it immediately but set it to run in the background click the **Schedule** button and select appropriate schedule type and settings. If the processing is scheduled for a later run you can use either [DB Job Expert](#) or DB Space Expert's built-in [Job Monitoring utility](#) to monitor and if necessary cancel scheduled jobs or kill already running jobs.

 **Tips:** Like many other utilities found in DB Space Expert the partition reorganization job can run in the foreground and in the background. For information on supported job and schedule types, advantages and disadvantages of different methods, job logging and auditing see [Running Space Reorganization Jobs](#) topic.

Converting Partitioned Tables to Regular Tables

1. In the [Tablespace Navigator](#) select any partition of an existing partitioned table you want to convert back to regular non-partitioned table.
2. Click the **Partitions** menu and then click the **Unpartition Table** command. Alternatively you can click the Unpartition  button on the application toolbar. This will open the **Convert Partitioned Table to Regular Table** dialog. An example screenshot of the dialog follows.




Convert Partitioned Table to Regular Table

Current Table

Owner: SCOTT Table: PART_TABLE_WTH_LOB
 Partition: P1 Tablespace: USERS2

Table Storage:

Tablespace: USERS 

Processing Options

Use PARALLEL option for faster processing
 (tables with object columns or LOB columns don't support this option; this option may also cause overallocated space for "small" objects)


Use UNRECOVERABLE and NOLOGGING options for faster processing

Rebuild global and local indexes after table is converted


Rebuild object statistics after the object is modified
 (run *smart* ANALYZE TABLE/INDEX commands after table is converted; this option increases processing time but ensures the database can use best query optimization plans)

Recompile dependent code objects
 (recompile dependent PROCEDURES, TRIGGERS, PACKAGES, etc..., after table is converted)


Convert Now Schedule... Cancel

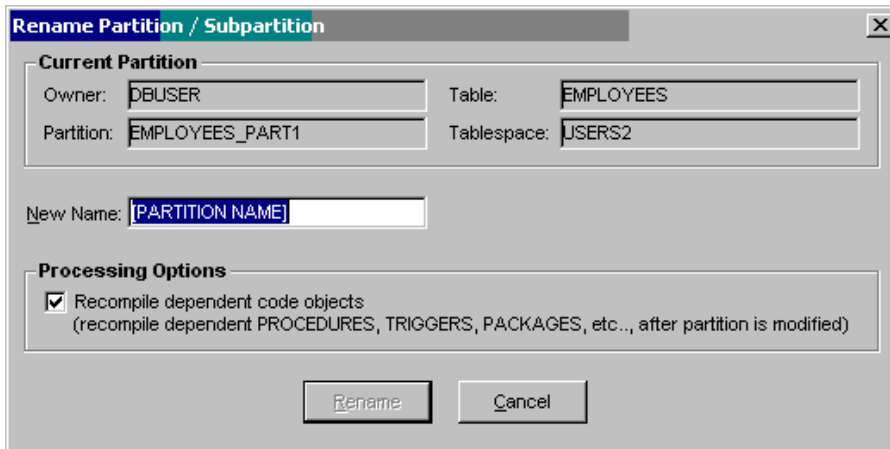
3. If you want to create the converted table in a different tablespace select new tablespace name and storage parameters. To edit storage parameters click the Storage  button.

- To start the conversion operation immediately on the front-end script, click the **Convert Now** button. To schedule this operation for a later run, perhaps off-hours, or to start it immediately but set it to run in the background click the **Schedule** button and select appropriate schedule type and settings. If the processing is scheduled for a later run you can use either [DB Job Expert](#) or DB Space Expert's built-in [Job Monitoring utility](#) to monitor and if necessary cancel scheduled jobs or kill already running jobs.

 **Tips:** Like many other utilities found in DB Space Expert the partition reorganization job can run in the foreground and in the background. For information on supported job and schedule types, advantages and disadvantages of different methods, job logging and auditing see [Running Space Reorganization Jobs](#) topic.


Renaming Partitions and Subpartitions

- In the [Tablespace Navigator](#) select table or index partition or subpartition you want to rename.
- Click the **Partitions** menu and then click the **Rename** command. Alternatively you can click the Rename  button on the application toolbar. This will open the **Rename Partition / Subpartition** dialog. An example screenshot of the dialog follows.



- Enter new name. This name must conform to Oracle object naming standards.
- Check the **Recompile Dependencies** checkbox if you want DB Space Expert to automatically recompile all table dependencies after the change. Click the **Rename** button to start the processing.

Splitting Partitions

- In the [Tablespace Navigator](#) select table partition you want to split to 2 or more smaller partitions.
- Click the **Partitions** menu and then click the **Split** command. Alternatively you can click the Split  button on the application toolbar. This will open the **Split Partition** dialog. An example screenshot of the dialog follows.

Split Partition [X]

Current Partition

Owner: SH Table: COSTS
 Partition: COSTS_Q1_1998 Tablespace: EXAMPLE
 Range: ('1998-04-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')

Number of New Partitions: 3 Partition Key Column(s):

New Partitions:

	Less Than Value	Partition Name	Tablespace
1.	TO_DATE('1998-02-01 00:00:00')	COSTS_JAN_1998	EXAMPLE
2.	TO_DATE('1998-03-01 00:00:00')	COSTS_FEB_1998	EXAMPLE
3.	TO_DATE('1998-04-01 00:00:00')	COSTS_MAR_1998	EXAMPLE

Processing Options

Use PARALLEL option for faster processing
 (tables with object columns or LOB columns don't support this option; this option may also cause overallocated space for "small" objects)

Use UNRECOVERABLE and NOLOGGING options for faster processing

Rebuild global and local indexes after partition is modified

Rebuild object statistics after the object is modified
 (run *smart* ANALYZE TABLE/INDEX commands after partition is modified; this option increases processing time but ensures the database can use best query optimization plans)

Recompile dependent code objects
 (recompile dependent PROCEDURES, TRIGGERS, PACKAGES, etc., after partition is modified)


Split Now Schedule... Cancel


- Enter **Number of New Partitions** value. You can type in the required number or you can use arrows up and down to increment/decrement the value as much as needed. The entered number must be greater or equal 2.
- Enter partition names, storage parameters and partition range/list specification into **Partitions** grid. Note that the dialog shows different partition properties for different partition types.

If **By Range** partition type is used then ranges are used for partitioning table data. Ranges must be defined for each partition using the values less than (value) clause, and the value specified in the cannot include a volatile data operation, such as calls to non-standard functions. String and date values must be enclosed in single quotes, as you would use them in other SQL expressions.

	Less Than Value	Partition Name	Tablespace
1.	TO_DATE('1998-02-01 00:00:00')	COSTS_JAN_1998	EXAMPLE
2.	TO_DATE('1998-03-01 00:00:00')	COSTS_FEB_1998	EXAMPLE
3.	TO_DATE('1998-04-01 00:00:00')	COSTS_MAR_1998	EXAMPLE

- Enter partition names and expressions for **less than (value)** clause. Partition names must conform to Oracle object naming standards.
- Select partition locations using the **Tablespace** drop-down list.
- Enter partition storage parameters in case if you do not want to use tablespace defaults.


To edit storage parameters click the Storage  button. You can select different tablespaces and storage parameters for different partitions.


 **Note:** Entered storage parameters will be ignored if the selected tablespace is a locally managed tablespace with automatic extent management.

If **By Hash** partition type is chosen then Oracle automatically calculates data ranges for data partitioning and you only enter partition names, locations and storage parameters.

	Partition Name	Tablespace
1.	P_1	USERS
2.	P_2	USERS
3.	P_3	EXAMPLE

- 4.1. Enter partition names. Partition names must conform to Oracle object naming standards.
- 4.2. Select partition locations using the **Tablespace** drop-down list
- 4.3. Enter partition storage parameters in case if you do not want to use tablespace defaults.


To edit storage parameters click the Storage  button. You can select different tablespaces and storage parameters for different partitions.


 **Note:** Entered storage parameters will be ignored if the selected tablespace is a locally managed tablespace with automatic extent management.

If **By List** partition type is chosen then you must enter list of key values for each selected partition.


	Values	Partition Name	Tablespace
1.	'AK', 'AL', 'AR', 'AZ'	P_A_STATES	USERS
2.	'CA', 'CO', 'CT'	P_C_STATES	USERS
3.	'DE', 'FL'	P_LESS_DE_FL	EXAMPLE

- 4.1. Enter partition names and list values. Partition names must conform to Oracle object naming standards.
- 4.2. Select partition locations using the **Tablespace** drop-down list.
- 4.3. Enter partition storage parameters in case if you do not want to use tablespace defaults.


To edit storage parameters click the Storage  button. You can select different tablespaces and storage parameters for different partitions.

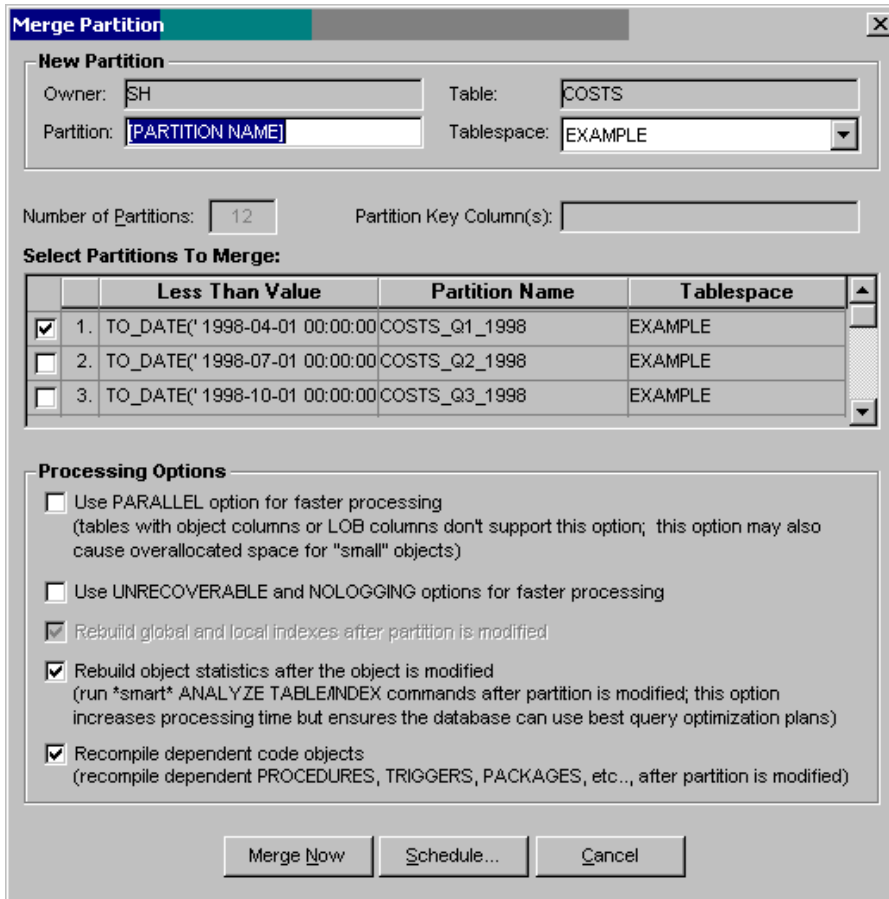
 **Note:** Entered storage parameters will be ignored if the selected tablespace is a locally managed tablespace with automatic extent management.

5. In the **Processing Options** box select advanced processing options for the conversion operation.
6. To start the splitting operation immediately on the front-end script, click the **Merge Now** button. To schedule this operation for a later run, perhaps off-hours, or to start it immediately but set it to run in the background click the **Schedule** button and select appropriate schedule type and settings. If the processing is scheduled for a later run you can use either [DB Job Expert](#) or DB Space Expert's built-in [Job Monitoring utility](#) to monitor and if necessary cancel scheduled jobs or kill already running jobs.

 **Tips:** Like many other utilities found in DB Space Expert the partition reorganization job can run in the foreground and in the background. For information on supported job and schedule types, advantages and disadvantages of different methods, job logging and auditing see [Running Space Reorganization Jobs](#) topic.

Merging Partitions

1. In the [Tablespace Navigator](#) select table partition of you want to merge with one or more other partitions creating a bigger partition.
2. Click the **Partitions** menu and then click the **Merge** command. Alternatively you can click the Split  button on the application toolbar. This will open the **Merge Partition** dialog. An example screenshot of the dialog follows.



Merge Partition

New Partition

Owner: SH Table: COSTS
 Partition: [PARTITION NAME] Tablespace: EXAMPLE

Number of Partitions: 12 Partition Key Column(s):

Select Partitions To Merge:

	Less Than Value	Partition Name	Tablespace
<input checked="" type="checkbox"/>	1. TO_DATE('1998-04-01 00:00:00	COSTS_Q1_1998	EXAMPLE
<input type="checkbox"/>	2. TO_DATE('1998-07-01 00:00:00	COSTS_Q2_1998	EXAMPLE
<input type="checkbox"/>	3. TO_DATE('1998-10-01 00:00:00	COSTS_Q3_1998	EXAMPLE

Processing Options

Use PARALLEL option for faster processing
 (tables with object columns or LOB columns don't support this option; this option may also cause overallocated space for "small" objects)

Use UNRECOVERABLE and NOLOGGING options for faster processing


Rebuild global and local indexes after partition is modified


Rebuild object statistics after the object is modified
 (run *smart* ANALYZE TABLE/INDEX commands after partition is modified; this option increases processing time but ensures the database can use best query optimization plans)

Recompile dependent code objects
 (recompile dependent PROCEDURES, TRIGGERS, PACKAGES, etc..., after partition is modified)


Merge Now Schedule... Cancel


3. Enter new partition name into the **Partition** field.
4. Use the **Tablespace** drop-down box to choose tablespace for the new merged partition. If you don't change the partition tablespace, by default the merged partition is created in the same tablespace and with the same storage parameters as the selected tablespace.
5. In the **Select Partitions to Merge** grid place a checkmark into the left most column for all partitions that you want to merge together. Note the partition selected in the Tablespace Navigator is checked by default.

 **Important Note:** Only adjacent partitions can be merged.
6. In the **Processing Options** box select advanced processing options for the merge operation.
7. To start the merging operation immediately on the front-end script, click the **Merge Now** button. To schedule this operation for a later run, perhaps off-hours, or to start it immediately but set it to run in the background click the **Schedule** button and select appropriate schedule type and settings. If the processing is scheduled for a later run you can use either [DB Job Expert](#) or DB Space Expert's built-in [Job Monitoring utility](#) to monitor and if necessary cancel scheduled jobs or kill already running jobs.


 **Tips:** Like many other utilities found in DB Space Expert the partition reorganization job can run in the foreground and in the background. For information on supported job and schedule types, advantages and disadvantages of different methods, job logging and auditing see [Running Space Reorganization Jobs](#) topic.

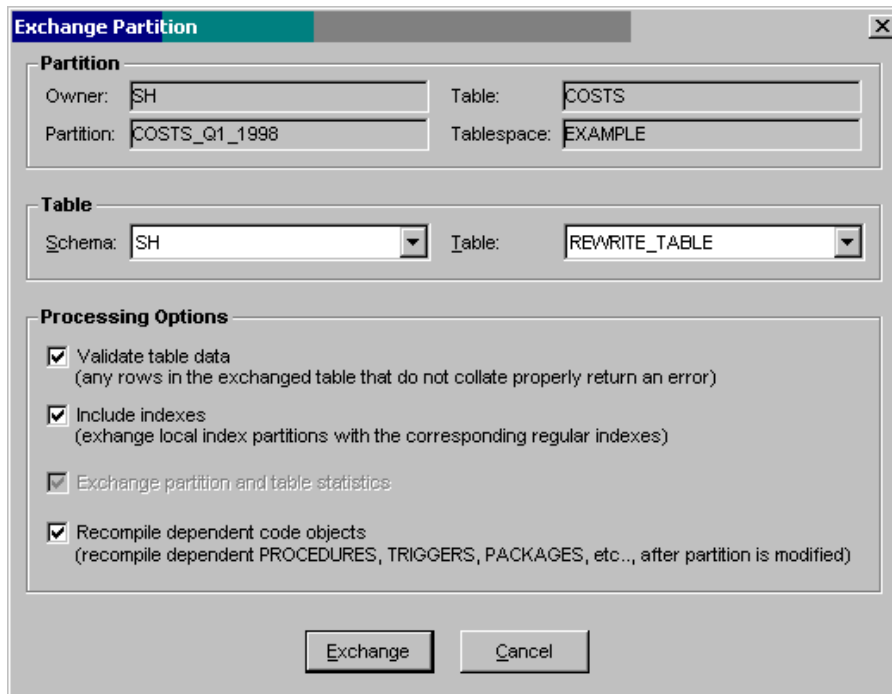
Exchanging Partition and Table Data

 **Background Information:** You can convert data in a partition into a non-partitioned table and also convert a non-partitioned table into a partition of a partitioned table by exchanging their data (and index) segments. Exchanging table partitions is most useful when you have an application using non-partitioned tables, which you want to convert to partitions of a partitioned table. For example, you may already have partition views that you wish to migrate into partitioned tables. This type of operation is much more efficient than INSERT...SELECT/DELETE method using intermediate table because the entire operation is performed on the segment level by updating references in the data dictionary tables.

 **Tip:** Do not confuse this type of data exchange operation with table and partition conversion utilities: [Converting Regular Tables to Tables with Partitions](#) and [Converting Partitioned Tables to Regular Tables](#)

Do the following steps to exchange partition data:


1. In the [Tablespace Navigator](#) select table partition you want to exchange.
2. Click the **Partitions** menu and then click the **Exchange** command. Alternatively you can click the Exchange  button on the application toolbar. This will open the **Exchange Partition** dialog. An example screenshot of the dialog follows.

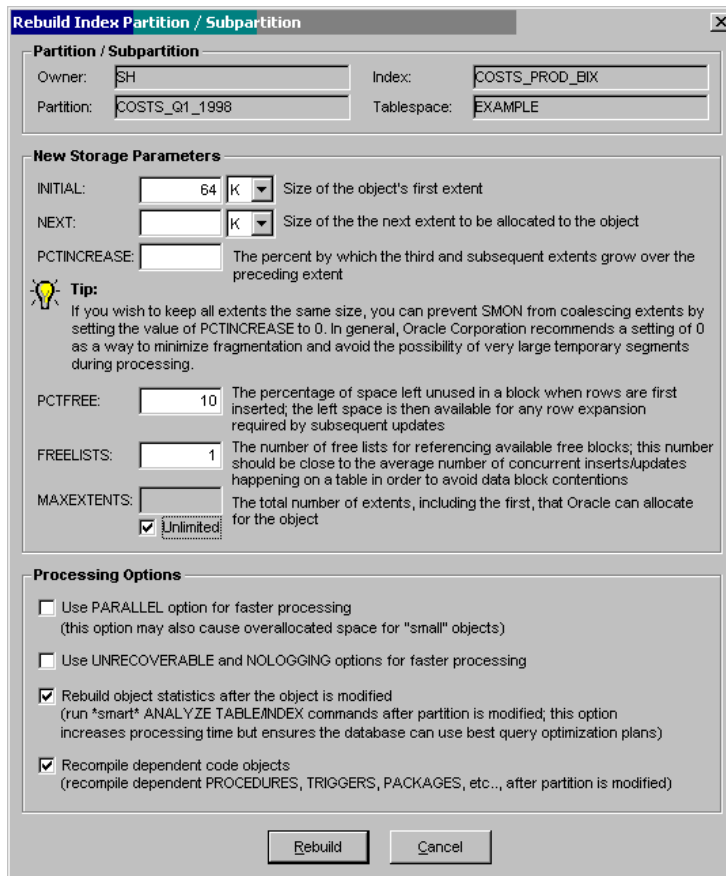


3. In the **Table** box select **Schema** name of the table to exchange partition data with.
4. In the **Table** box select **Table** name of the table to exchange partition data with.
5. If you want to validate table data in order to ensure it matches the partition range/list clause and will not create any problems later check the **Validate table data** checkbox
6. If you want to automatically exchange table and partition indexes check the Include indexes checkbox. If you do not check that option affected table and partition indexes will be marked UNUSABLE.

7. Check the **Recompile dependent code objects** option if you want DB Space Expert to automatically recompile all table and partition dependencies after the exchange operation is complete.
8. Click the **Exchange** button to start the processing.

Rebuilding Index Partitions and Subpartition

1. In the [Tablespace Navigator](#) select index partition you want to rebuild.
2. Click the **Partitions** menu and then click the **Rebuild** command. Alternatively you can click the Rebuild  button on the application toolbar. This will open the **Rebuild Partition** dialog. An example screenshot of the dialog follows.



Rebuild Index Partition / Subpartition

Partition / Subpartition

Owner: SH Index: COSTS_PROD_BIX
 Partition: COSTS_Q1_1998 Tablespace: EXAMPLE

New Storage Parameters

INITIAL: 64 K Size of the object's first extent
 NEXT: K Size of the the next extent to be allocated to the object
 PCTINCREASE: The percent by which the third and subsequent extents grow over the preceding extent

Tip:
 If you wish to keep all extents the same size, you can prevent SMON from coalescing extents by setting the value of PCTINCREASE to 0. In general, Oracle Corporation recommends a setting of 0 as a way to minimize fragmentation and avoid the possibility of very large temporary segments during processing.

PCTFREE: 10 The percentage of space left unused in a block when rows are first inserted; the left space is then available for any row expansion required by subsequent updates
 FREELISTS: 1 The number of free lists for referencing available free blocks; this number should be close to the average number of concurrent inserts/updates happening on a table in order to avoid data block contentions
 MAXEXTENTS: Unlimited The total number of extents, including the first, that Oracle can allocate for the object

Processing Options

Use PARALLEL option for faster processing
 (this option may also cause overallocated space for "small" objects)

Use UNRECOVERABLE and NOLOGGING options for faster processing

Rebuild object statistics after the object is modified
 (run *smart* ANALYZE TABLE/INDEX commands after partition is modified; this option increases processing time but ensures the database can use best query optimization plans)

Recompile dependent code objects
 (recompile dependent PROCEDURES, TRIGGERS, PACKAGES, etc., after partition is modified)

Rebuild Cancel

3. Fill in **New Storage Parameters** for the selected partition. Description of each parameter and recommendations are provided directly on the dialog.
4. In the **Processing Options** box select advanced processing options for the rebuild operation.
5. Click the **Rebuild** button to start the rebuild operation.

Rebuilding Unusable Indexes

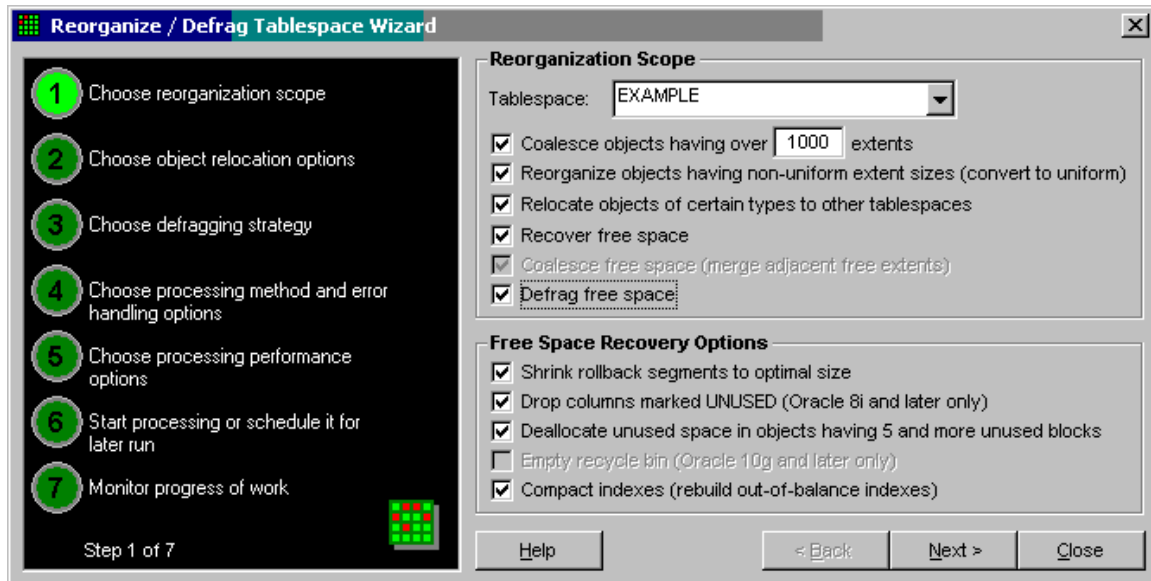
1. In the [Tablespace Navigator](#) select table partition whose local unusable indexes you want to fix.
2. Click the **Partitions** menu and then click the **Rebuild Unusable Indexes** command. DB Space Expert will start rebuilding unusable indexes.

- A status message will be displayed as soon as the processing is complete.

Reorganize / Defrag Tablespace Wizard

DB Space Expert supports a number of different tablespace reorganization, optimization and defragging options. All supported option use **on-line reorganization methods** using standard Oracle commands and can be performed in the background transparently to running database applications. In certain cases you may experience measurable performance impact when rebuilding large indexes and tables. To simply the reorganization DB Space supports flexible scheduling methods that allow scheduling the reorganization process to run during off-peak times.



To access these reorganization options use the **Reorganize / Defrag Tablespace Wizard**. This Wizard can be started using **Tools/Reorganize / Defrag Tablespace Wizard** menu or alternatively by clicking on the Reorg  button on the application toolbar.



Top-level reorganization options

The following table describes supported top-level reorganization options and their usage instructions

Option	Description
Reorganization Options	
Coalesce objects having lots of extents	Use this option to reorganize tablespace objects whose number of extents exceeds certain threshold. DB Space Expert can rebuild such objects with new optimized storage parameters ensuring the rebuilt object consists of no more than few extents. When calculating new storage parameters DB Space Expert takes into account tablespace storage parameters as well as average sizes of other objects. In non-uniform size tablespaces it calculates uniform size extents that can be easily reused by other objects in case if the reorganized object is dropped or moved to another tablespace.
Reorganize objects having non-uniform extent	In order to eliminate tablespace fragmentation caused by different extent sizes DR Space Expert can reorganize existing objects and modify their


sizes	extent sizes making all extents uniform or proportional to the uniform size. In tablespaces with non-uniform extent allocation methods it calculates average objects sizes to figure out most common extent size and then finds nearest common extent size value proportional to the size of the database data block.
Relocate objects of certain types to other tablespaces	Use option to quickly move all tables or all LOBs or all indexes or any combination of these objects to different tablespaces. You can pick up to 3 different tablespaces at a time. For example, you can use this option to quickly separate tables and table indexes into separate tablespaces as well as to separate LOB columns containing movies, graphics or large documents of other types.
Recover free space	This option allows you to reclaim free tablespace space. Use the following Free Space Recovery Options to choose which space recovery methods to employ during the reorganization process.
Coalesce free space (merge adjacent free extents)	This option is automatic and used internally by the Wizard after each reorganization step. For more information about free space coalesce process see Coalescing Tablespace Free Space topic.
Defrag free space	Use this option to merge non-adjacent free space extents. DB Space Expert supports several methods for performing this type reorganization. See the following paragraphs for more details
Free Space Recovery Options	
Shrink rollback segments to optimal size	Check this option to allow DB Space Expert to shrink large rollback segments to their optimal sizes. Please note that rollback segment optimal sizes are calculated by the Oracle database. DB Space Expert simply executes the appropriate ALTER ROLLBACK SEGMENT commands.
Drop columns marked UNUSED	Check this option to allow DB Space Expert to permanently drop unused table columns, in other words columns marked by users as unneeded.  Note: This option is only enabled in Oracle databases version 8i and up.
Deallocate unused space in objects having 5 or more unused blocks.	Check this option to allow DB Space Expert to run the Deallocate Unused Space operation as part of the reorganization process.
Empty recycle bin	Check this option to allow DB Space Expert to automatically empty the tablespace Recycle Bin .  Note: This option is only enabled in Oracle databases version 10g and up.
Compact indexes (rebuild out-of-balance indexes)	Check this option to allow DB Space Expert to find out-of-balance indexes and rebuild them.

Batch Methods for Separating Objects of Different Types

If you choose the **Relocate objects of certain types** option on the first dialog screen of the **Reorganize / Defrag Tablespace Wizard**, on the second step you will be then presented with another dialog screen that you can use to select which types you want to relocate and their target tablespaces.

Separate set of options is available for the following 3 groups of object types:

- Tables, table partitions and subpartitions
- Indexes, index partitions and subpartitions
- LOBs, LOB partitions and subpartitions.

 **Note:** LOB option applies to all types of LOBs. LOB indexes are always relocated together with LOBs segments. Oracle currently does not support storing LOB segments and indexes in different tablespaces.

For each group of objects you can pick either **Leave** or **Relocate** option. If you pick the Leave option, DB Space Expert will do nothing to that group of objects, in terms of the object relocation.


Objects Defragging and Free Space Defragging Strategies.

The **Reorganize / Defrag Tablespace Wizard** currently supports 3 defragging strategies that can be employed to defrag fragmented objects and free space extents. In all cases to implement the on-line application-transparent reorganization DB Space Expert uses some other tablespaces as scratch pad to move database segments from and to the tablespace being reorganized. It can use either an existing tablespace or create new tablespace, which it will automatically drop after the reorganization processing is successfully completed.

Important Notes:

- If you choose to use an existing tablespace as a scratch pad make sure to pick a PERMANENT type tablespaces. If you pick a TEMPORARY tablespace and the database is shutdown in a middle of the reorganization process you may loose affected tablespace objects and data.
- Because all reorganizations are performed within the database using a scratch tablespace, the selected scratch tablespace must have sufficient free space to hold all the objects being reorganized. If the selected scratch tablespace has sufficient free space but AUTOEXTENSIBLE option turned on for the tablespace files, Oracle will automatically expand this database as needed. However after the reorganization is completed such tablespace will not be shrunk back to its pre-reorganization size. That's why it is highly recommended to allow DB Space Expert temporarily create a new scratch tablespace that it can drop after the processing. DB Space Expert by default uses DEFrag as the new scratch tablespace name.


The following table describes supported defragging strategies.

Strategy	Description
Automatic	Choose object's processing order based on the analysis of the free space allocation within the tablespace and the type and number of tablespace segments and extents. Reorganize free space by moving out objects separating chunks of free space, then merging adjacent free extents and then defragging other objects which will fill the coalesced free space. Last, move back previously moved out objects.  Tip: This strategy might be unavailable in situations when a number of different reorganization options have been selected on the first page. In such situations DB Space Expert could be unable to make correct prognosis of space allocation in the tablespace and calculate the best defragging path.
Small objects first	First, free space in the tablespace by moving out small objects then defrag large objects and then move back small objects filling any remaining gaps.
Large objects first	First, free space in the tablespace by moving out largest objects then coalesce free space, rebuild and defrag small objects, and then move back previously moved large objects.

Error Handling Methods

The following 3 methods are supported:

Method	Description
Abort after error	This method can be used with both foreground and background reorganization jobs. This method instructs DB Space Expert to abort the processing after any error.
Ignore errors	This method can be used with both foreground and background reorganization jobs. This method instructs DB Space Expert to ignore processing error and continue with the next object. It is usually safe to use this method. In large tablespaces with lots of objects and method some kind of a minor error or warning is quite likely. In all cases all processing errors are logged to the screen or to the ORA_MONITOR.DB_SPACE_LOG log table
Prompt me what to do	This method can be used with foreground reorganization jobs only. This method instructs DB Space Expert to pause the processing after any error, display an interactive message containing the error details and name of the affected object or storage structure. You can then choose to ignore the error and continue, abort the processing completely or retry the last step and continue. In the latest case you can correct the error cause and pick the retry to Retry.


 **Tip:** If an error occurs in background space reorganization job, DB Space Expert will attempt to use [DB Alert Expert](#) to email or page the error. If DB Alert Expert is not installed in the database or DB Alert Console is not running at the time of the error, the email/page notification will fail and DB Space Expert will continue ignoring this failure.

Advanced Processing Options

DB Space Expert supports a number of options that allow you to get maximum possible performance out of your tablespace reorganization jobs and ensure that objects and storage structures being reorganized do not affect existing database application.

Processing Options

- Whenever possible move objects ONLINE
(this option requires more processing time and space)
- Use PARALLEL option for faster processing
(tables with object columns or LOB columns don't support this option;
this option may also cause overallocated space for "small" objects)
- Use UNRECOVERABLE and NOLOGGING options for faster processing
- Rebuild indexes after moving tables and table partitions
(this option increases processing time but ensures that indexes remain intact)
- Rebuild statistics after moving objects
(run *smart* ANALYZE TABLE/ANALYZE INDEX commands after
objects are moved or rebuilt; this option increases processing time but
ensures the database can use best query optimization plans)
- Recompile dependent code objects
(recompile dependent PROCEDURES, TRIGGERS, PACKAGES, etc.,
after the object is moved or rebuilt)

 **Important Notes:** These processing options are applied for each object during or immediately after that object processing to ensure that existing database applications persistently remain in a usable state with as little impact on the running applications as possible.

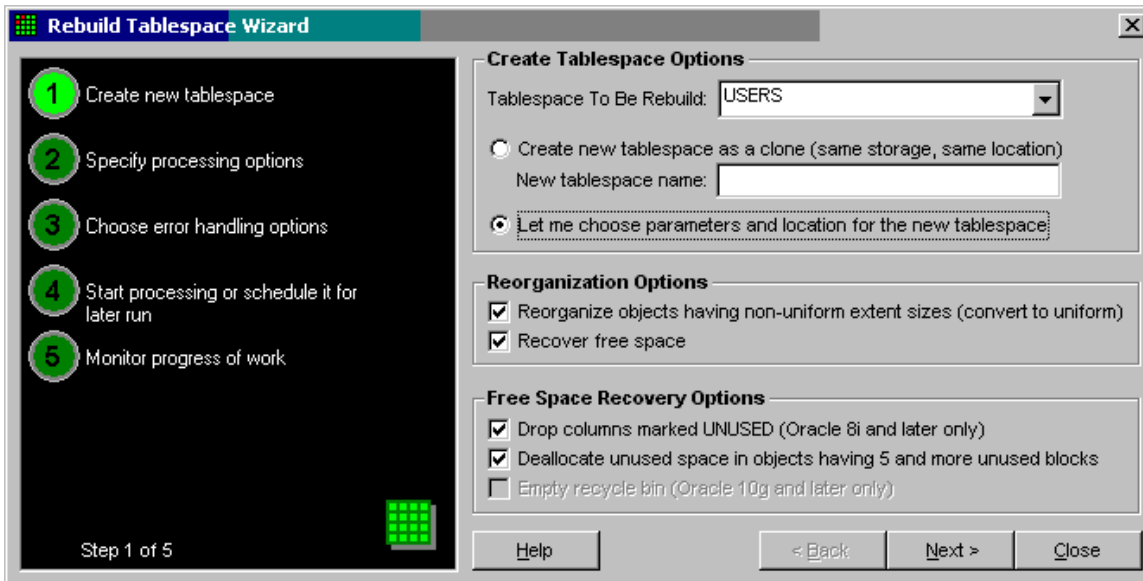
Reorganization Job Scheduling

Like many other utilities found in DB Space Expert the **Reorganize / Defrag Tablespace Wizard** can run tablespace reorganization jobs in the foreground and in the background. For information on supported job and schedule types, advantages and disadvantages of different methods, job logging and auditing see [Running Space Reorganization Jobs](#) topic.

Rebuild Tablespace Wizard

Rebuilding tablespace is bar the fastest and most efficient method to defragment a tablespace as well as fix many other space allocation issues. DB Space Expert provides special utility Rebuild Tablespace Wizard that can be used to rebuild and clone tablespaces. This utility allows you with just a few mouse clicks to build a new tablespace with the same or different storage parameters as the tablespace being rebuilt and then move all existing tablespace objects to the new tablespace. **Because all objects are moved sequentially and created with uniform extent sizes and zero PCT_INCREASE the new tablespace has zero fragmentation and ensures that free space and object fragmentation don't become an issue again some time later.**

This Wizard can be started using **Tools/Rebuild Tablespace Wizard** menu or alternatively by clicking on the Rebuild  button on the application toolbar.





Top-level reorganization options

The following table describes supported top-level tablespace rebuild options and their usage instructions

Create tablespace Options

Option	Description
Create tablespace options	
Create new tablespace as a clone (same storage	Use this option to clone the selected tablespace. The new tablespace will be created with the same number of files in the same directory as the existing


same location)	tablespace. You also need to specify new tablespace name. To ensure uniqueness of new file names DB Space will generate new file names as copies of old file names with the date suffix inserted before the file extension.
Let me choose parameters and location of the new tablespace.	Use this option if you want to specify new tablespace location, file names and storage parameters. If the option is selected the Wizard will popup the Create Tablespace dialog , which you can use to create the new tablespace.
Tablespace Reorganization Options	
Reorganize objects having non-uniform extent sizes	Use this option if you are building new tablespace without using Oracle automatic extent management features (Oracle 8i and up). If this option is selected DB Space Expert will migrate all objects to the new tablespace and reorganize their storage parameters to enforce uniform extent sizing. This will prevent future fragmentation problems in the new tablespace.
Recover free space	This option allows you to reorganize existing objects by freeing their unused space, which in turn will reduce space requirements for the new tablespace. Use the following Free Space Recovery Options to choose which space recovery methods to employ during the reorganization process.
Free space recovery options	
Drop columns marked UNUSED	Check this option to allow DB Space Expert to permanently drop unused table columns, in other words columns marked by users as unneeded.  Note: This option is only enabled in Oracle databases version 8i and up.
Deallocate unused space in objects having 5 or more unused blocks.	Check this option to allow DB Space Expert to run the Deallocate Unused Space operation as part of the reorganization process.
Empty recycle bin	Check this option to allow DB Space Expert to automatically empty the tablespace Recycle Bin .  Note: This option is only enabled in Oracle databases version 10g and up.

Error Handling Methods

The following 3 methods are supported:

Method	Description
Abort after error	This method can be used with both foreground and background reorganization jobs. This method instructs DB Space Expert to abort the processing after any error.
Ignore errors	This method can be used with both foreground and background reorganization jobs. This method instructs DB Space Expert to ignore processing error and continue with the next object. It is usually safe to use this method. In large tablespaces with lots of objects and method some kind of a minor error or warning is quite likely. In all cases all processing errors are logged to the screen or to the ORA_MONITOR.DB_SPACE_LOG log table
Prompt me what to do	This method can be used with foreground reorganization jobs only. This method instructs DB Space Expert to pause the processing after any error, display an interactive message containing the error details and name of the affected object or

	storage structure. You can then choose to ignore the error and continue, abort the processing completely or retry the last step and continue. In the latest case you can correct the error cause and pick the retry to Retry.
--	---


 **Tip:** If an error occurs in background space reorganization job, DB Space Expert will attempt to use [DB Alert Expert](#) to email or page the error. If DB Alert Expert is not installed in the database or DB Alert Console is not running at the time of the error, the email/page notification will fail and DB Space Expert will continue ignoring this failure.

Advanced Processing Options

DB Space Expert supports a number of options that allow you to get maximum possible performance out of your tablespace reorganization jobs and ensure that objects and storage structures being reorganized do not affect existing database application.

Processing Options

- Whenever possible move objects ONLINE
(this option requires more processing time and space)
- Use PARALLEL option for faster processing
(tables with object columns or LOB columns don't support this option;
this option may also cause overallocated space for "small" objects)
- Use UNRECOVERABLE and NOLOGGING options for faster processing
- Rebuild indexes after moving tables and table partitions
(this option increases processing time but ensures that indexes remain intact)
- Rebuild statistics after moving objects
(run *smart* ANALYZE TABLE/ANALYZE INDEX commands after
objects are moved or rebuilt; this option increases processing time but
ensures the database can use best query optimization plans)
- Recompile dependent code objects
(recompile dependent PROCEDURES, TRIGGERS, PACKAGES, etc.,
after the object is moved or rebuilt)


 **Important Notes:** These processing options are applied for each object during or immediately after that object processing to ensure that existing database applications persistently remain in a usable state with as little impact on the running applications as possible.

Reorganization Job Scheduling

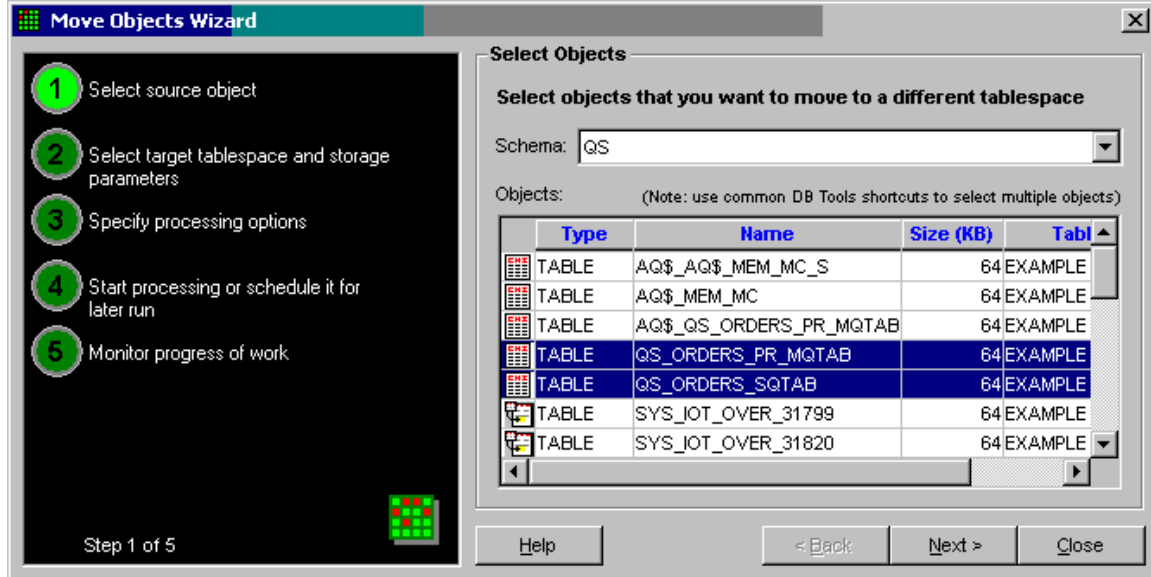
Like many other utilities found in DB Space Expert the **Rebuild Tablespace Wizard** can run tablespace reorganization jobs in the foreground and in the background. For information on supported job and schedule types, advantages and disadvantages of different methods, job logging and auditing see [Running Space Reorganization Jobs](#) topic.

Move Objects Wizard

The Move Objects Wizard can be used to move single or multiple objects between tablespaces.

This Wizard can be started using **Tools/Move Objects Wizard** menu or alternatively by clicking on the Move Object  button on the application toolbar. It can be also started using right-click menu in the

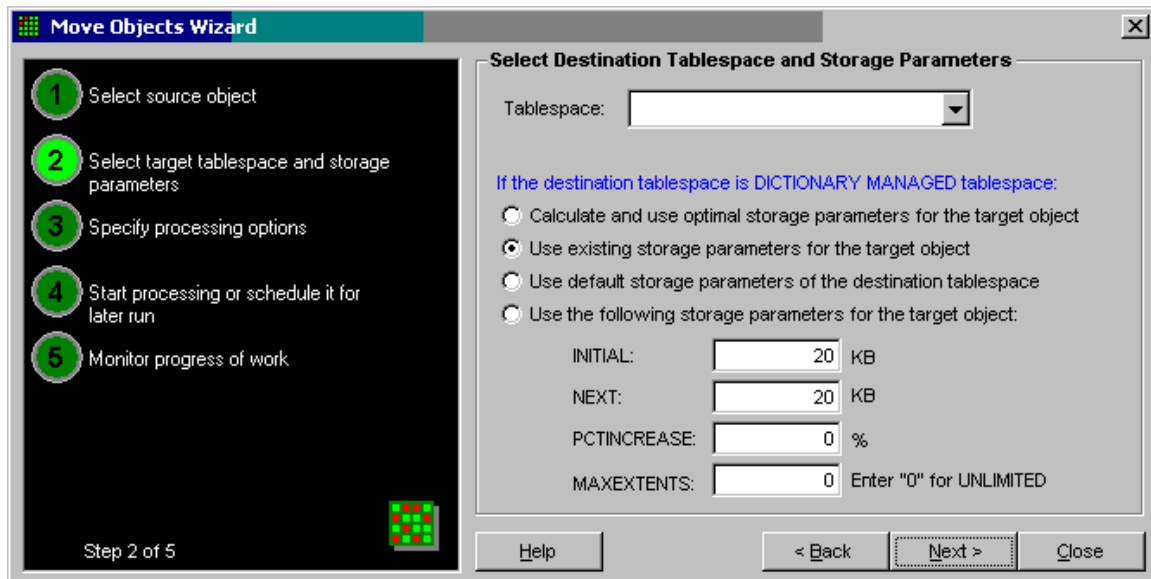
[Tablespace Navigator](#). In the latest case the Wizard will automatically load optimal settings and name of the schema selected in the Tablespace Navigator.



Using Move Objects Wizard

On the first screen of the Move Objects Wizard you can select one or more objects that you want to move. This can be accomplished in 2 steps:

1. In the **Schema** drop-down list select name of the schema whose objects you are going to move. The Move Objects Wizard will populate the **Objects** grid with all storage objects from the selected schema.
2. Using common DB Tools shortcuts select all required objects. The description of supported shortcuts is available in the [Select Tables](#) topic. Click the **Next** button to advance to the next screen.



3. Select destination tables from the **Tablespace** drop-down list.
4. Choose storage parameters for target objects. Please note if you are moving multiple objects the selected parameters will be applied to all objects. Click the **Next** button to advance to the next screen.

5. Specify advanced processing options. Note that DB Space Expert supports a number of options that allow you to get maximum possible performance out of your object reorganization jobs and ensure that objects and storage structures being reorganized do not affect existing database application.

Processing Options

Whenever possible move objects ONLINE
(this option requires more processing time and space)


Use PARALLEL option for faster processing
(tables with object columns or LOB columns don't support this option;
this option may also cause overallocated space for "small" objects)

Use UNRECOVERABLE and NOLOGGING options for faster processing

Rebuild indexes after moving tables and table partitions
(this option increases processing time but ensures that indexes remain intact)

Rebuild statistics after moving objects
(run *smart* ANALYZE TABLE/ANALYZE INDEX commands after
objects are moved or rebuilt; this option increases processing time but
ensures the database can use best query optimization plans)

Recompile dependent code objects
(recompile dependent PROCEDURES, TRIGGERS, PACKAGES, etc...,
after the object is moved or rebuilt)

 **Important Notes:** These processing options are applied for each object during or immediately after that object processing to ensure that existing database applications persistently remain in a usable state with as little impact on the running applications as possible.


6. Choose how and when you want to run this object reorganization job. You can choose to run it immediately as a foreground script or schedule it for background processing.

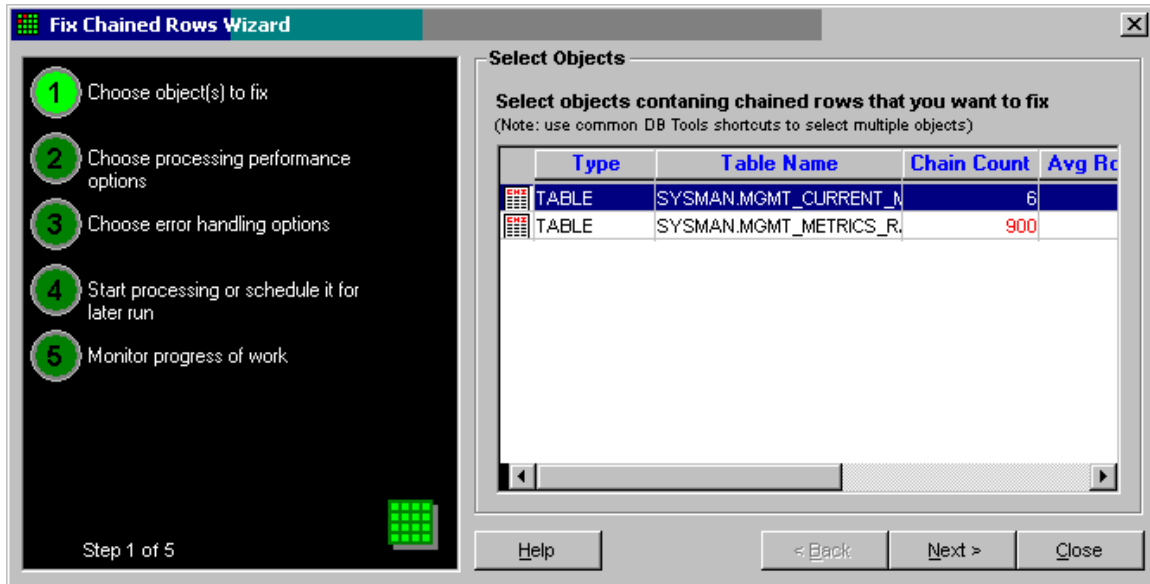
Reorganization Job Scheduling

Like many other utilities found in DB Space Expert the **Move Objects Wizard** can run object reorganization jobs in the foreground and in the background. For information on supported job and schedule types, advantages and disadvantages of different methods, job logging and auditing see [Running Space Reorganization Jobs](#) topic.

Fix Chained Rows Wizard

The **Fix Chained Rows Wizard** utility can be used correct space allocation problems caused by chained and migrated rows problems. Best of all you can use this utility to correct all chained rows in any number of tables with just a few mouse clicks.

This Wizard can be started using **Tools/Fix Chained Rows Wizard** menu or alternatively by clicking on the Move Fix Chains  button on the application toolbar.

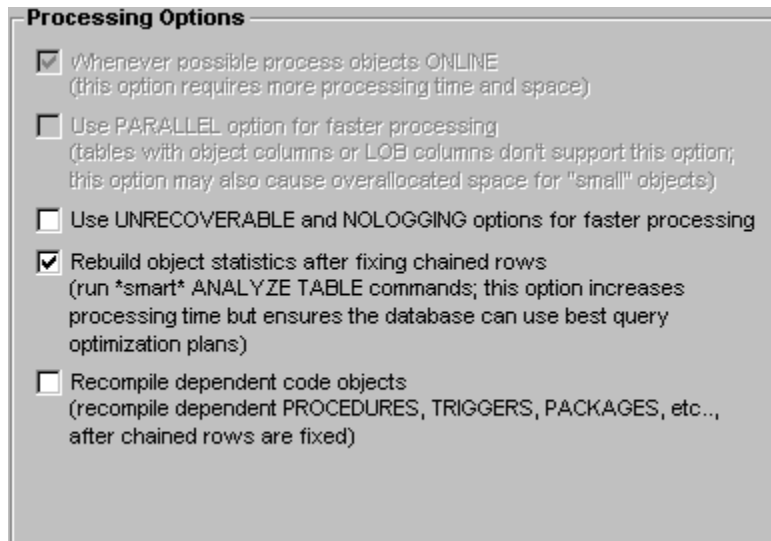



Using Fix Chained Rows Wizard

1. On the first screen the Fix Chained Rows Wizard lists all database tables, partitions and subpartitions currently experiencing chained rows problem.

Select one or more objects that you want to fix. Use common DB Tools shortcuts to select multiple objects. The description of supported shortcuts is available in the [Select Tables](#) topic. Click the **Next** button to advance to the next screen.

2. Specify advanced processing options. Note that DB Space Expert supports a number of options that allow you to get maximum possible performance out of your object reorganization jobs and ensure that objects and storage structures being reorganized do not affect existing database application.



 **Important Notes:** These processing options are applied for each object during or immediately after that object processing to ensure that existing database applications persistently remain in a usable state with as little impact on the running applications as possible.

3. Choose how and when you want to run this object reorganization job. You can choose to run it immediately as a foreground script or schedule it for background processing.

Reorganization Job Scheduling

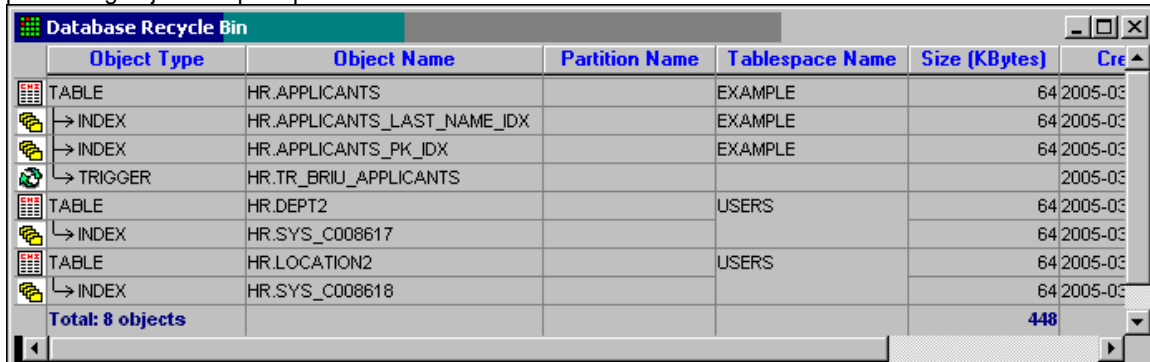
Like many other utilities found in DB Space Expert the **Fix Chained Rows Wizard** can run object reorganization jobs in the foreground and in the background. For information on supported job and schedule types, advantages and disadvantages of different methods, job logging and auditing see [Running Space Reorganization Jobs](#) topic.

Working with Database Recycle Bin

Oracle 10g features special storage area in every tablespace called the Recycle Bin where dropped objects are retained until the object is either recovered via a Flashback Drop operation, or until the object is purged from the Recycle Bin. DB Space Expert includes the **Recycle Bin Explorer** utility that can be used to visually browse database recycle bin contents and purge recycled objects as needed.

To start the **Recycle Bin Explorer** use **Tools/Explore Recycle Bin** menu or click the Recycle Bin  button on the application toolbar.

For your convenience the Recycle Bin Explorer presents all dropped objects in a tree like structure presenting object "drop" dependencies.



	Object Type	Object Name	Partition Name	Tablespace Name	Size (KBytes)	Cre
	TABLE	HR.APPLICANTS		EXAMPLE	64	2005-03
	→ INDEX	HR.APPLICANTS_LAST_NAME_IDX		EXAMPLE	64	2005-03
	→ INDEX	HR.APPLICANTS_PK_IDX		EXAMPLE	64	2005-03
	→ TRIGGER	HR.TR_BRIU_APPLICANTS				2005-03
	TABLE	HR.DEPT2		USERS	64	2005-03
	→ INDEX	HR.SYS_C008617			64	2005-03
	TABLE	HR.LOCATION2		USERS	64	2005-03
	→ INDEX	HR.SYS_C008618			64	2005-03
	Total: 8 objects				448	

This presentation method is self-explaining. You can easily see, for example, that the trigger HR.TR_BRIU_APPLICANTS was dropped when its parent table HR.APPLICANTS was dropped.

The Recycle Bin Explorer also displays other useful information including dropped objects name, tablespace name, object size in Kbytes, time object has been created and also time object has been dropped.

To purge objects from the recycle bin and as a result permanently delete them from the database and free their space use any of the following methods:

To purge a single particular object:

1. Select the object you want to purge.
2. Select **File/Purge Selection** menu. DB Space Expert will purge the selected object.

To purge a group of objects:

1. Use common DB Tools shortcuts to select multiple objects. The description of supported shortcuts is available in the [Select Tables](#) topic.
2. Select **File/Purge Selection** menu. DB Space Expert will purge all selected objects.

To purge a group of objects:

1. Select **File/Purge All** menu. DB Space Expert will purge all objects in the database recycle bin.

Running Space Reorganization Jobs

Job Execution Methods

DB Space Expert supports 3 different methods of executing various space reorganization jobs. You can pick different method in different situation or stick with a single method and uses for all reorganizations.

1. **Foreground Jobs** – These jobs are executed on the front-end running DB Space Expert graphical Interface. DB Space Expert executes job script command-by-command. After each command it evaluation the command completion status and based on the returned value decides what to do next. As the job progresses DB Space Expert displays job progress of work messages on the console. Only one foreground job can run at a time. However you can open multiple DB Space Expert instances to run multiple foreground jobs simultaneously.
2. **Background Immediate Job** - These jobs are executed on the back-end using Oracle native scheduling API available in the DBMS_JOB package. If the reorganization script is small (less then 2000) bytes DB Space Expert submits it as a simple job. If the script is large it creates one or more temporary procedures and schedules the entry-point procedure starting immediately after job submission. The created procedures are then automatically deleted at the end of the processing.

The Background Immediate Jobs use DBMS_PIPE package to communicate with the DB Space Expert front-end console and report their progress of work messages and errors. For every immediate job DB Space Expert opens a separate graphical log window, which displays job progress of work messages and errors sent using DBMS_PIPE package functions. All displayed messages are sorted by processing time. You can change the default sorting if required. You can also filter, print, zoom and resize logs as needed. Note that for your convenience all error messages are displayed in red. Multiple background immediate jobs can be started and monitored simultaneously.



Important Notes: DB Space Experts console cannot be closed until all background immediate jobs complete. Restarting your computer or forcibly terminating DB Space Expert using Windows Task Manager while background immediate jobs are in progress can cause background jobs to hang while communicating with the non-existing console.

Background Scheduled Jobs - These jobs are executed on the back-end using Oracle native scheduling API available in the DBMS_JOB package. If the reorganization script is small (less then 2000) bytes DB Space Expert submits it as a simple job. If the script is large it creates one or more temporary procedures and schedules the entry-point procedure starting at the user-specified time. The created procedures are then automatically deleted at the end of the processing.

If an error occurs in the Background Scheduled Job, job scripts will attempt to use DB Alert Expert interface to notify you about the error. If DB Alert Expert is not installed in the database or DB Alert Console is not running at the time of the error, the email/page notification will fail without affecting the rest of the job script. Similarly on the job successful completion the job will attempt to notify you will attempt to use DB Alert Expert interface to notify you about job completion status.



Important Notes: The Background Scheduled Jobs use ORA_MONITOR.DB_SPACE_LOG table to log their progress of work messages. If the ORA_MONITOR user or DB_SPACE_LOG table do not exist DB Space Expert will automatically create them with your permission. You can use DB Tools SQL Editor or any other tool of your choice to periodically poll messages in the ORA_MONITOR.DB_SPACE_LOG table.

Job Logging

Job logging is automatic and always on. You do not need to do anything to enable this feature.

Different types of reorganization job use different logging methods. Read the previous topic for more information about different jobs types and the implementation of their logging methods.

Advantages and disadvantages of different reorganization job types

The main 2 advantages of the **Foreground Job** type are:

- The option to use interactive error handling method. In case of an error an interactive message is displayed on the screen giving you a choice to ignore the error and continue processing, to abort the processing or to intervene and then restart the processing from the last failed operation.
- The processing can be easily interrupted or paused at any time by simply clicking on the Cancel button.

The main disadvantages of the **Foreground Job** type are

- The DB Space Expert graphical interface cannot be closed while the job is running and cannot be used for other tasks.

The main advantages of the **Background Immediate Job** type are:

- Multiple jobs can be started and run on the server simultaneously.
- Easy to read graphical interface is available for job logging. Progress of work messages appear on the screen in a real-time.

The main disadvantages of the **Background Immediate Job** type are:

- The DB Space Expert graphical interface cannot be closed while such jobs are running. However the interface can be used for other tasks.
- Background jobs cannot be paused or interacted with. The only way to stop such job is to kill it using DB Space Expert's built in [job monitoring and management utility](#) or using DB Job Expert component.

The main advantages of the **Background Scheduled Job** type are:

- Multiple jobs can be started and run on the server simultaneously.

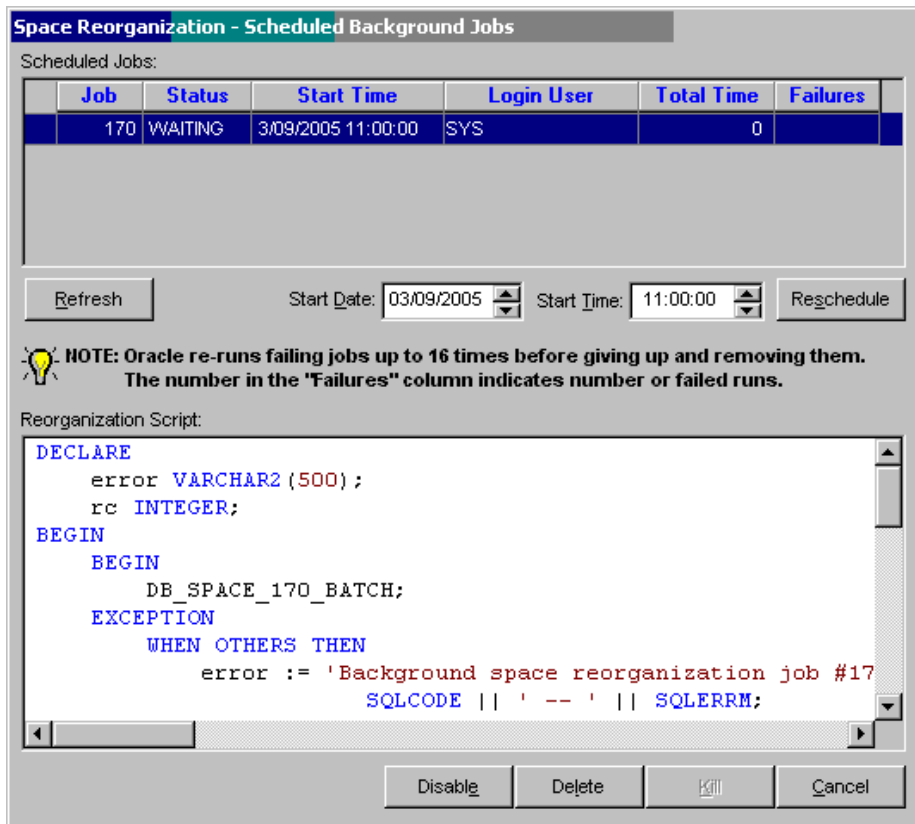
The main disadvantages of the **Background Immediate Job** type are:


- Background jobs cannot be paused or interacted with. The only way to stop such job is to kill it using DB Space Expert's built in [job monitoring and management utility](#) or using DB Job Expert component.
- This job requires existence of the ORA_MONITOR.DB_SPACE_LOG table

Monitoring and Managing Space Reorganization Jobs

DB Space Expert features a built-in scheduling interface that is optimized specifically for the space reorganization jobs. This scheduling interface uses Oracle DBMS_JOB package to submit and modify reorganization jobs as well as monitor their statuses.



Here is a sample screenshot of the built-in scheduling interface.



To monitor already submitted jobs and reschedule, disable, delete or kill running jobs use **Scheduled Background Jobs** dialog that can be started from the **Tools/Scheduled Background Jobs** menu or alternately clicking the Jobs  button on the application toolbar

The following controls are available on this dialog

Control	Description
Scheduled Jobs grid	This control displays all pending and running space reorganization jobs scheduled by you and by other users.
Refresh button	Use this button to refreshes contents of the Scheduled Jobs grid.
Start Date edit field, Start Time edit field and Reschedule button	These 3 controls can be used to reschedule a pending job to run at a different time. The action applies to the currently selected job in the Scheduled Jobs grid.
Reorganization Script display control	This read-only control displays the definition of the reorganization job selected in the Scheduled Jobs grid. In most cases you will only see the a call to the "entry-point" reorganization procedure and the error handling block attempting to invoke DB Alert Expert 's procedure for email notification in case of a job error and on successful job completions.
Disable/Enable button	This control can be used to toggle enabled state of a pending job. The action applies to the currently selected job in the Scheduled Jobs grid.

	<p>grid.</p> <p> Tips:</p> <ul style="list-style-type: none"> You can use this button temporarily disable jobs in order to prevent their running. You can enable jobs when it is necessary to you or delete them if you don't need them anymore. Disabling already running jobs does not stop them. Use the Kill button instead and then disable them.
Delete button	Use this button to delete a pending job. This button cannot be used to stop a running job. The action applies to the currently selected job in the Scheduled Jobs grid.
Kill button	<p>Use this button to kill the database session of the running job. The action applies to the currently selected job in the Scheduled Jobs grid.</p> <p> Important Notes: Killing a job is a dangerous operation which should be used as a last resort only. Killing a job will leave you with an incomplete object and/or space reorganization changes and also possible leave in the database a bunch of intermediate procedures generated by DB Space Expert for the reorganization job.</p>
Close button	This button simply closes the dialog. Closing the dialog does not affect any scheduled jobs.

Reports

Tablespace Space Usage

This report displays printable graph of tablespace free and used space allocation and other important tablespace statistics.

Database Files

This report displays all database files and their properties

Object Summary

This report displays summary counts for different types of database objects.


Fragmentation

This reports displays all fragmented database objects that have four or more extents. For your convenience, the report data is grouped:

- Group by scheme
- Subgroup by object (segment)


Tablespace Trends

This report displays various trend graphs and data for a given tablespace. Most interesting are space allocation and fragmentation trending graphs that can provide with a clear idea of where your tablespaces are heading and help you with planning your future space requirements as well as help you to predict future problems proactively.

 **Important Notes:** This report uses historical trend data collected by the [DB Trends Expert](#) component. The DB Trends weekly statistics collection job must be setup in the database and running for at least a few weeks before you can obtain any meaningful trend information.

Database Trends

This report displays various trend graphs and data for the entire database. Most interesting are space allocation and fragmentation trending graphs that can provide with a clear idea of where your database is heading and help you with planning your future space requirements as well as help you to predict future problems proactively.

 **Important Notes:** This report uses historical trend data collected by the [DB Trends Expert](#) component. The DB Trends weekly statistics collection job must be setup in the database and running for at least a few weeks before you can obtain any meaningful trend information.

DB Consumer Expert

Overview

DB Consumer Expert is a one-stop solution for managing your Oracle database users, security settings, database resource quotas and resource usage and priority plans.

It provides easy to use graphical interface to the Oracle Database Resource Manager that allows Oracle DBA to manage database server resources effectively. Resources can be managed and distributed differently to different users and user groups as well as they can be adjusted differently during different resource-demand periods.

DB Consumer Expert also extends Oracle's native resource usage functions found in the Database Resource Manager with custom extensions that greatly expand the resource usage scheduling options.

In addition it provides a large set of analytical reports for exploring database usage patterns and trends by database users and database applications. Using these reports you can easily pinpoint who is consuming most of your database resources and which applications are used. You can then use the available DB Consumer Expert's resource management functions to adjust database resource quotas and priorities thus ensuring that most critical business applications get what they need while other applications continue to operate with acceptable response time levels.



Important Note: Resource-management parts of DB Consumer Expert require Oracle database version 8i or later. User, Role, Space Quota, Security management parts can be used with any Oracle database version 7.3 and up.



For your information:

Oracle Database Resource Manager is an obscure tool that became available in Oracle 8i. However, it received little attention because the resource management interface requires deep knowledge of the Oracle database and complex instructions for interacting with the Database Resource Manager. Normally adding a simple directive to an existing resource usage plan requires executing a number of stored procedures, which can be hardly done without any programming. DB Consumer Expert closes that gap by providing simple and easy to use graphical interface where all functions can be performed with just a few mouse clicks.


Oracle Database Resource Manager consists of 3 basic components:

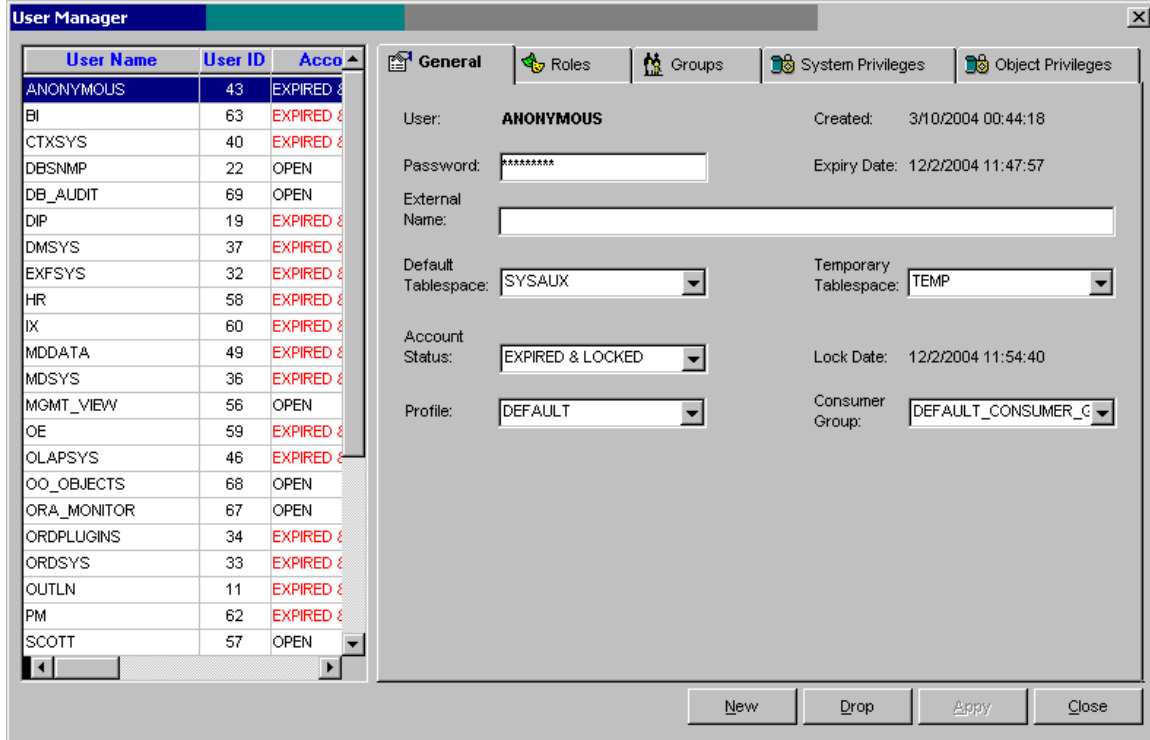
Resource Consumer Groups - A resource consumer group is a collection of users with similar requirements for resource consumption. Users can be assigned to more than one resource consumer group, but each user's active session can only be assigned to one resource consumer group at a time. For example, different consumer groups can be setup for OLTP users, for DSS users and for ad hoc queries.

Resource Plans - A resource plan describes the resources allocated to one or more resource consumer group(s). For example, the amount of CPU utilization permitted for a group of users, the total amount of undo space allowed for the use by the entire group, and so on.

Resource Plan Directives - Resource plan directives allocate resources among the resource consumer groups in the resource plan. Essentially, directives connect resource consumer groups to their resource plans. Resource plan directives use resource allocation methods to determine how the Database Resource Manager will allocate resources to a resource consumer group or resource plan.

Managing Database Users

To start the User Manager click **Manage/Manage Users** menu or click the Users  button on the application toolbar. The **User Manager** dialog will appear.



The User Manager dialog consists of 2 parts: The left hand side is used to list all database users and their properties in a single table-like list. The right hand side is used to display and edit user properties assign roles and assign various system-level and object-level privileges. The settings and properties displayed on the right hand side are associated with the user whose name is selected in the user list on the left hand side.

You can use standard DB Tools list sorting options to rearrange how user names or properties appear in the list. For more information about supported sorting options read the [Sorting](#) topic.

User Properties

Different Oracle database versions support different sets of user properties. On the **General** tab page DB Consumer Expert displays only options supported in the database you are connected to.

User – Specify Oracle user account name.

Lock Date – Date and time when the user account was created. This property is automatic and cannot be changed.

Password – Specify user password. Use this property when creating Oracle users identified locally by the database, in other words, to indicate that the user must specify password to log on to the database

External Name – Specify External Name to create an external user. Such a user must be authenticated by an external service, such as an operating system or a third-party service. In this case, Oracle database relies on the login authentication of the operating system to ensure that a specific operating system user has access to a specific database user.



Important Note: Oracle strongly recommends that you do not use external authentication with operating systems that have inherently weak login security. For more information, see Oracle Database Administrator's Guide.

Default Tablespace – Specify the default tablespace for objects that the user creates. If you omit this property, then the user's objects are stored in the database default tablespace. If no default tablespace has been specified for the database, then the user's objects are stored in the SYSTEM tablespace.



Restrictions: You cannot specify a locally managed tablespace, including an undo tablespace, or a dictionary-managed temporary tablespace as a user's default tablespace.

Temporary Tablespace – Specify the tablespace for the user's temporary segments. If you omit this property, then the user's temporary segments are stored in the database default temporary tablespace or, if none has been specified, in the SYSTEM tablespace.



Restrictions: The specified tablespace must be a temporary tablespace and must have a standard block size. The tablespace cannot be an undo tablespace or a tablespace with automatic segment-space management

Account Status – Specify user account status, one of the following:

- **OPEN** – user account is active and enabled
- **LOCKED** – user account is inactive and disabled.
- **EXPIRED** – user account is enabled but the password is expired; user must change his/her password on the next logon to the database
- **EXPIRED & LOCKED** – user account is inactive and disabled; in addition user password is expired. If the account status later changes to **OPEN** the password expiry status remains in affect.

Expiry Date – Date and time when the user password expired last time (expired accounts) or will expire (open accounts).

Lock Date – Date and time when the user account was locked last time. This property is automatic and cannot be changed.

Profile – Specify the default profile you want to assign to the user. The profile limits the amount of database resources the user can use. If you omit this property, then Oracle database assigns the DEFAULT profile to the user.

Consumer Group – Specify the default consumer group you want to assign to the user. If Oracle [Database Resource Manager](#) is not activated this property has no effect.

See your Oracle documentation for additional information about user account properties and how to use them.

Enabling and Disabling User Accounts

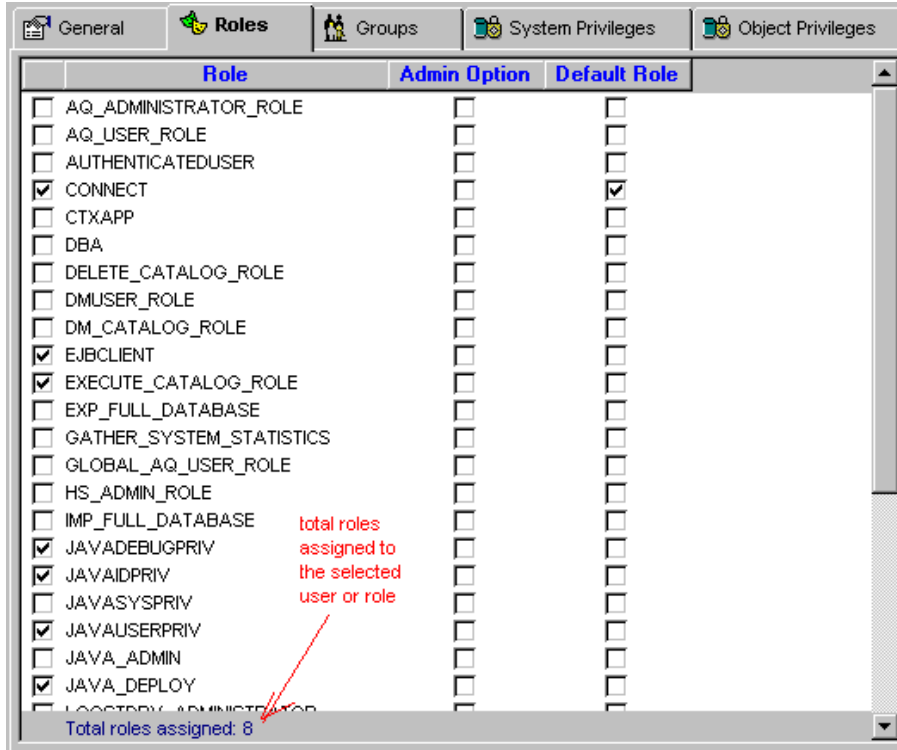
You can use the **General** page to change user account status. Select LOCKED or EXPIRED & LOCKED status from **Account Status** drop-down list to disable the user account. Select OPEN status to re-enable the user account. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

Forcing Users to Change Their Passwords

You can use the **General** page to change user account status. Select EXPIRED status from **Account Status** drop-down list to expire user password. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

Granting and Revoking Roles

The **Roles** tab page on the User Manager dialog can be used to control which roles are assigned to the selected user.



This is a multiple-choice screen. You can select as many roles as you want. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

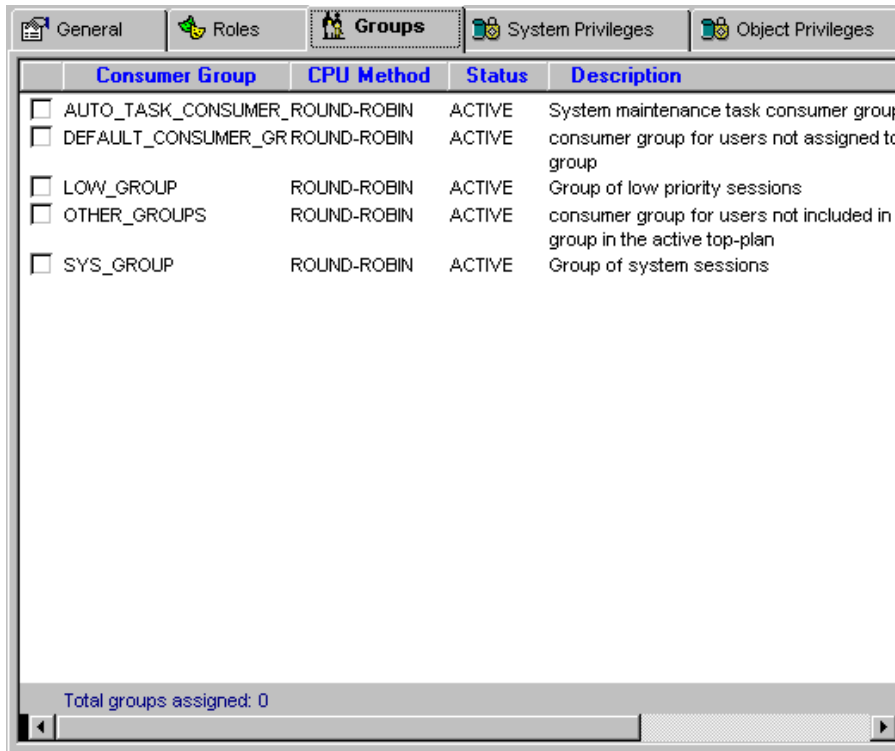
To grant a role to the selected user place a checkmark in the left-most column. To revoke a role from the selected user, clear the corresponding checkmark.

Checking the **Admin Option** box enables the user to grant this role to someone else. Checking the **Default Role** option instructs Oracle to automatically give the selected user access to the role during on connect even if the role is password-protected and normally requires the user to executing SET ROLE command in order to enable such role for the current session.

The **total** field shows how many roles are checked on the screen.


Granting and Revoking Consumer Groups

The **Groups** tab page on the User Manager dialog can be used to control which consumer groups are enabled for the selected user, in other words which groups are granted to the selected user.



This is a multiple-choice screen. You can select as many groups as you want. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

To grant a group to the selected user place a checkmark in the left-most column. To revoke a group from the selected user, clear the corresponding checkmark.

 **Tip:** An assigning consumer group to a user is not the same as using it. It simply grants the user privileges to switch to and use the assigned group whenever that user wants. Keep in mind that every database session can be switched to only one consumer group at a time. Yet there exists a number of reasons why may want to assign multiple groups to a single user. For example, you may want to allow the user (or database applications running under that user account) to switch to different groups and to use different resource usage plans during different times, use "LOW" usage plans for daily report processing and as opposite use "HIGH" usage plans for other mission critical business processing.

The **total** field shows how many groups are checked on the screen.

Granting and Revoking System Privileges

The **System Privileges** tab page on the User Manager dialog can be used to control which system privileges are enabled for the selected user, in other words which privileges are granted to the selected user.

Category	Privilege	Description
TABLESPACES	<input type="checkbox"/> MANAGE TABLESPACE	Take tablespaces off tablespace backups
	<input checked="" type="checkbox"/> UNLIMITED TABLESPACE	Use an unlimited amount of space. This privilege overrides any specific tablespace privilege from a user, further tablespace all specific tablespace privilege to roles.
TRIGGERS	<input type="checkbox"/> ADMINISTER DATABASE TRIGGER	Create a trigger on a database
	<input type="checkbox"/> ALTER ANY TRIGGER	Enable, disable, or compile a trigger
	<input type="checkbox"/> CREATE ANY TRIGGER	Create database triggers
	<input checked="" type="checkbox"/> CREATE TRIGGER	Create a database trigger
TYPES	<input type="checkbox"/> DROP ANY TRIGGER	Drop database triggers
	<input type="checkbox"/> ALTER ANY TYPE	Alter object types in a schema
	<input type="checkbox"/> CREATE ANY TYPE	Create object types in a schema
	<input checked="" type="checkbox"/> CREATE TYPE	Create object types in a schema
	<input type="checkbox"/> DROP ANY TYPE	Drop object types in a schema

Total privileges granted: 5

This is a multiple-choice screen. You can select as many privileges as you want. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

For your convenience all system privileged are grouped and sorted by category and comments explaining each privilege displayed in the right-most column.

To grant a privilege to the selected user place a checkmark in the left-most column. To revoke a privilege from the selected user, clear the corresponding checkmark.

The **total** field shows how many privileges are checked on the screen.

Granting and Revoking Object Privileges

The **Object Privileges** tab page on the User Manager dialog can be used to control which database object access privileges are enabled for the selected user, in other words which privileges are granted to the selected user.

Select Objects and Privileges

Owner: Tips: Click on a button to grant object privilege. To restrict privileges to certain columns only select them first.

Objects:

Object Type	Object Name
VIEW	EMP_DETAILS_VIEW
TABLE	COUNTRIES
TABLE	DEPARTMENTS
TABLE	EMPLOYEES
TABLE	JOBS
TABLE	JOB_HISTORY
TABLE	LOCATIONS
TABLE	REGIONS
SEQUENCE	DEPARTMENTS_SEQ
SEQUENCE	EMPLOYEES_SEQ

SELECT
INSERT
UPDATE
DELETE
READ
ALTER
REFERENCES
INDEX
EXECUTE

Admin Option - allow grantee to grant object privileges to other users and roles.

Table/View Columns:

Column Name
COUNTRY_ID
COUNTRY_NAME
REGION_ID

Granted Privileges:

	Privilege	Object Type	Owner	Object Name	Gra
<input checked="" type="checkbox"/>	EXECUTE	PACKAGE	SYS	DBMS_STATS	SYS
<input checked="" type="checkbox"/>	SELECT	TABLE	HR	COUNTRIES	SYSTEM
<input checked="" type="checkbox"/>	SELECT	TABLE	HR	DEPARTMENTS	SYSTEM
<input checked="" type="checkbox"/>	SELECT	TABLE	HR	EMPLOYEES	SYSTEM

This is a multiple-choice screen. You can select as many privileges as you want. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

The **Object Privileges** tab page is split into 2 logically separate parts. The top part - **Select Objects and Privileges** is used to grant new privileges. The bottom part - **Granted Privileges** displays all object-level privileges that have been already granted to the selected user and it can be also used to revoke these privileges.

To grant a privilege to the selected user:

1. Select object's owner name (schema name) in the **Owner** drop-down list. DB Consumer Expert will populate the **Objects** list with all grantable objects in specified schema name.
2. Select the object you want to grant to the user in the **Objects** list. If you select a table or view DB Consumer Expert will populate the **Table/View Columns** list displayed on the right hand side with names of all columns available in the selected object. It will also enable/disable "grant" buttons displayed in the middle of the screen accordingly to what privileges can be granted for the selected object type.
3. if necessary check the **Admin Option** box. Checking the **Admin Option** box enables the user to grant the given privilege to someone else.
4. Click the appropriate "grant" buttons (SELECT, INSERT, and so on) to grant specific privileges to the user. These privileges will be automatically appended to the end of the **Granted Privileges** grid. In case if you made a mistake simply uncheck the left-most column in the grid for the wrong privilege.

For UPDATE, INSERT and REFERENCES operations if required you can restrict user's access to specific table/view columns only. Select all required columns in the column list then click UPDATE or INSERT or REFERENCES button once. To select multiple columns hold down the CTRL key while clicking on column names. DB Consumer Expert grants all-columns access to the user when no columns are selected in the column list.

To revoke a privilege from the selected user, clear the corresponding checkmark on the **Granted Privileges** grid.

Creating New Database Users

1. Open **User Manager** dialog using **Manage/Manage Users** menu.
2. Click the **New** button.
3. Fill in new user properties on the **General** tab page.
4. Enter additional options and grant permissions using other tab pages. For detailed information on how to use other tab pages read previous topics in this chapter.
5. Click the **Apply** button to save changes.



Tip: After you have the new user created use [Space Usage Quotas](#) and [Resource Usage Quotas](#) tools to complete the user setup.


Deleting Database Users

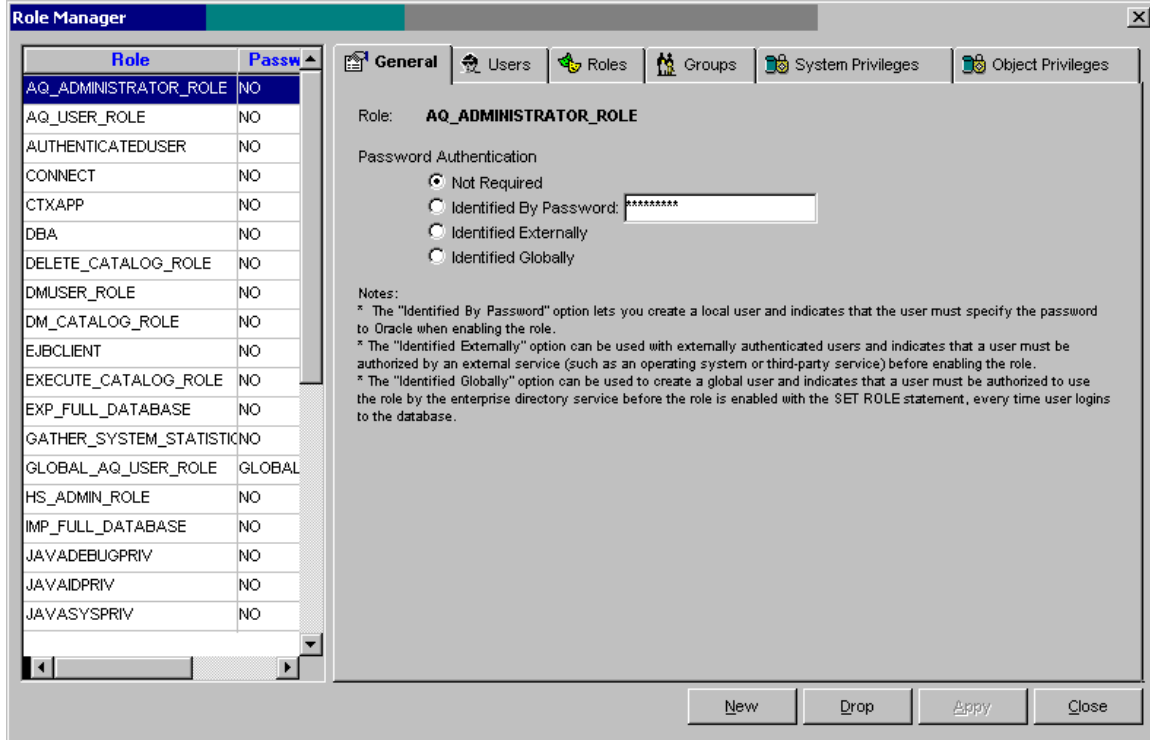
1. Open **User Manager** dialog using **Manage/Manage Users** menu.
2. On the left hand side of the dialog select name of the user you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the YES button if you are sure that the right user is chosen for deletion.



Important Note: DB Consumer Expert will automatically delete all user objects associated with the user schema as well as drop all referential constraints if there are any. This operation cannot be undone without restoring the user account, objects and data from a database backup.

Managing Database Roles

To start the Role Manager click **Manage/Manage Roles** menu or click the Roles  button on the application toolbar. The **Role Manager** dialog will appear.



The Role Manager dialog consists of 2 parts: The left hand side is used to list all database roles and their properties in a single table-like list. The right hand side is used to display and edit role properties assign roles to users and also assign various system-level and object-level privileges to roles. The settings and properties displayed on the right hand side are associated with the role whose name is selected in the role list on the left hand side.

You can use standard DB Tools list sorting options to rearrange how user names or properties appear in the list. For more information about supported sorting options read the [Sorting](#) topic.

Role Properties

Different Oracle database versions support different sets of role properties. On the **General** tab page DB Consumer Expert displays only options supported in the database you are connected to.

Role – Specify role name.

Password Authentication – Use these options to specify role password authentication.

Not Required – Choose this option to indicate that this role is authorized by the database and that no password is required to enable the role.

Identified by Password – Choose this option to create a local role and indicate that the user must specify the password to the database when enabling the role. The password can contain only single-byte characters from your database character set regardless of whether this character set also contains multi-byte characters.

Identified Externally – Choose this option to indicate that a user must be authenticated by an external service and then the user must use SET ROLE statement in order to enable this role.

Identified Globally – Choose this option to indicate that a user must be authenticated by a global enterprise directory service and then the user must use SET ROLE statement in order to enable this role.

See your Oracle documentation for more information about role properties and how to use them.

Granting and Revoking Roles to Users

The **Users** tab page on the Role Manager dialog can be used to control which users have been granted the selected role. This page is similar in functionality to the Granting and Revoking Roles page on the User Manager dialog, it is just features the opposite view of "Users granted the selected role" vs. "Roles granted to the selected user." Read [Granting and Revoking Roles](#) topic for instructions on how to use this screen.

Granting and Revoking Roles to Roles

The **Roles** tab page on the Role Manager dialog can be used to control which roles in effect have been granted the selected role. This page is similar in functionality to the Granting and Revoking Roles page on the User Manager dialog. Read [Granting and Revoking Roles](#) topic for instructions on how to use this screen.

Granting and Revoking Consumer Groups

The **Groups** tab page on the Role Manager dialog can be used to control which consumer groups are enabled for the selected role, in other words which groups are granted to the selected role.

This page is similar in functionality to the Granting and Revoking Consumer Groups page on the User Manager dialog." Read [Granting and Revoking Consume Groups](#) topic for instructions on how to use this screen.



Tips: An assigning of consumer group to a role is not the same as using it. It simply grants a privilege to switch to and use the assigned group whenever a user wants. Keep in mind that every database session can be switched to only one consumer group at a time. Yet there exists a number of reasons why may want to assign multiple groups to a single user. For example, you may want to allow the user (or database applications running under that user account) to switch to different groups and to use different resource usage plans during different times, use "LOW" usage plans for daily report processing and as opposite use "HIGH" usage plans for other mission critical business processing.

Granting and Revoking System Privileges

The **System Privileges** tab page on the Role Manager dialog can be used to control which system privileges are enabled for the selected role, in other words which privileges are granted to the selected role.

This page is similar in functionality to the Granting and Revoking System Privileges page on the User Manager dialog." Read [Granting and Revoking System Privileges](#) topic for instructions on how to use this screen.

Granting and Revoking Object Privileges

The **Object Privileges** tab page on the Role Manager dialog can be used to control which object-level privileges are enabled for the selected role, in other words which privileges are granted to the selected role.

This page is similar in functionality to the Granting and Revoking Object Privileges page on the User Manager dialog." Read [Granting and Revoking Object Privileges](#) topic for instructions on how to use this screen.

Creating New Database Roles

1. Open **Role Manager** dialog using **Manage/Manage Roles** menu.
2. Click the **New** button.
3. Fill in new role properties on the **General** tab page.
4. Enter additional options and grant permissions using other tab pages. For detailed information on how to use other tab pages read previous topics in this chapter.
5. Click the **Apply** button to save changes.


Deleting Database Roles

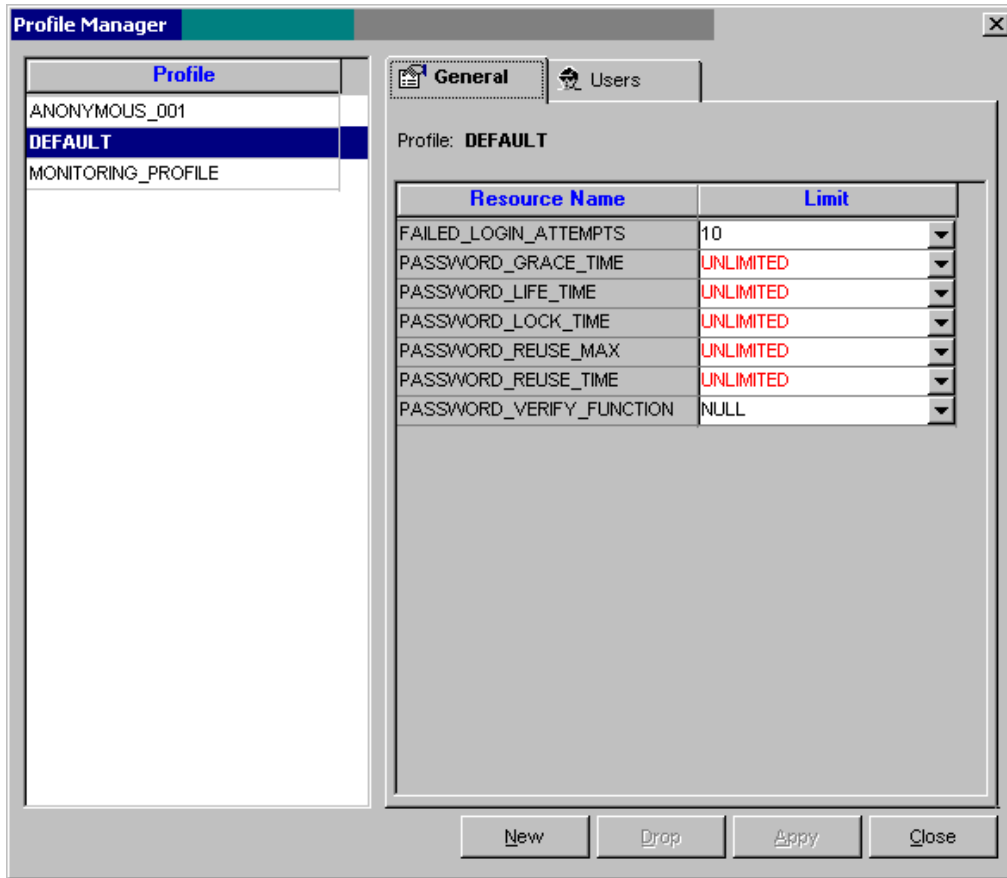
1. Open **Role Manager** dialog using **Manage/Manage Roles** menu.
2. On the left hand side of the dialog select name of the role you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the YES button if you are sure that the right role is chosen for deletion.



Important Note: All database users who have been assigned this role will automatically loose all privileges associated with this role.

Managing Database Profiles

To start the Profile Manager click **Manage/Manage Profiles** menu or click the Profiles  button on the application toolbar. The **Profile Manager** dialog will appear.



The Profile Manager dialog consists of 2 parts: The list box on the left hand side lists all database profiles. The right hand side of the dialog screen is used to display and edit profile properties and also used to assign profiles to users. The settings and properties displayed on the right hand side are associated with the profile whose name is selected in the profile list on the left hand side.

You can use standard DB Tools list sorting options to rearrange how profile names appear in the list. For more information about supported sorting options read the [Sorting](#) topic.



Tips:

- Only security related settings are displayed in profile properties. To manage resource usage-related properties use [Resource Usage Quotas](#) dialog.
- Different profile properties are supported in different Oracle versions. DB Consumer Expert displays only properties supported in the database you are connected to.
- Consult your Oracle documentation for detailed information about supported profile properties and their values.

Profile Properties

Database Security

FAILED_LOGIN_ATTEMPTS - Specify the number of failed attempts to log in to the user account before the account is locked.

Password Security

Parameters that set lengths of time are interpreted in number of days. For testing purposes you can specify minutes or even seconds as a decimal number. For example, to specify 5 minutes enter 0.003472 (this value was calculated $1/24/60*5$).

PASSWORD_LIFE_TIME- Specify the number of days the same password can be used for authentication. If you also set a value for **PASSWORD_GRACE_TIME**, the password expires if it is not changed within the grace period, and further connections are rejected. If you do not set a value for **PASSWORD_GRACE_TIME**, its default of **UNLIMITED** will cause the database to issue a warning but let the user continue to connect indefinitely.

PASSWORD_REUSE_TIME and **PASSWORD_REUSE_MAX** - These two parameters must be set in conjunction with each other. **PASSWORD_REUSE_TIME** specifies the number of days before which a password cannot be reused. **PASSWORD_REUSE_MAX** specifies the number of password changes required before the current password can be reused. For these parameter to have any effect, you must specify an integer for both of them.

If you specify an integer for both of these parameters, then the user cannot reuse a password until the password has been changed the password the number of times specified for **PASSWORD_REUSE_MAX** during the number of days specified for **PASSWORD_REUSE_TIME**.

For example, if you specify **PASSWORD_REUSE_TIME** to 30 and **PASSWORD_REUSE_MAX** to 10, then the user can reuse the password after 30 days if the password has already been changed 10 times.

If you specify an integer for either of these parameters and specify **UNLIMITED** for the other, then the user can never reuse a password.

If you specify **DEFAULT** for either parameter, then Oracle Database uses the value defined in the **DEFAULT** profile. By default, all parameters are set to **UNLIMITED** in the **DEFAULT** profile. If you have not changed the default setting of **UNLIMITED** in the **DEFAULT** profile, then the database treats the value for that parameter as **UNLIMITED**.

If you set both of these parameters to **UNLIMITED**, then the database ignores both of them.

PASSWORD_LOCK_TIME - Specify the number of days an account will be locked after the specified number of consecutive failed login attempts.

PASSWORD_GRACE_TIME - Specify the number of days after the grace period begins during which a warning is issued and login is allowed. If the password is not changed during the grace period, the password expires.

PASSWORD_VERIFY_FUNCTION - Specify the name of the password complexity verification routine. Oracle database provides a default script, but you can create your own routine instead. Specify **NULL** to indicate that no password verification is performed.

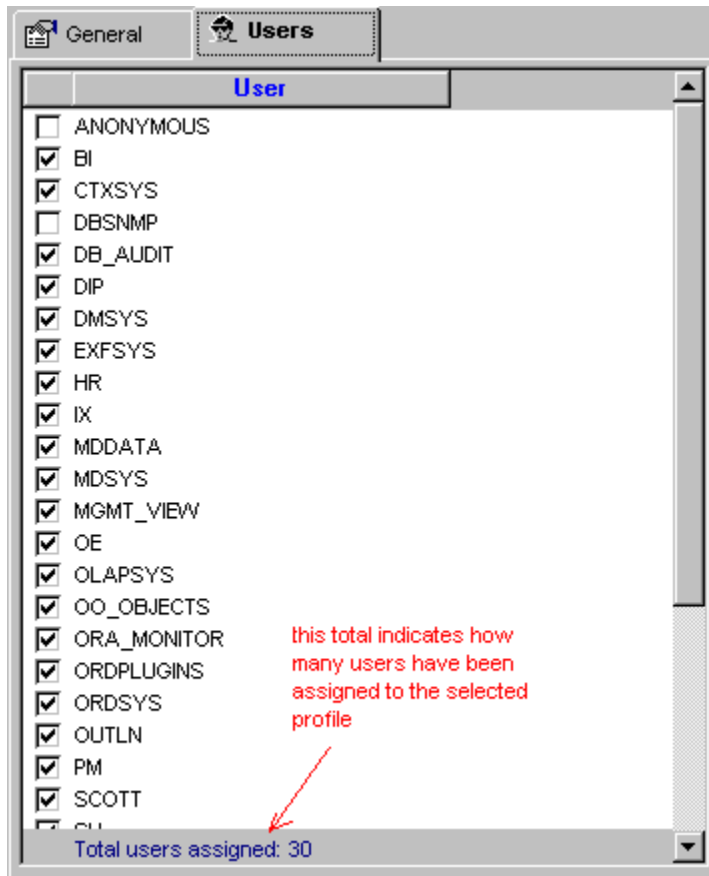


Tip: **PASSWORD_VERIFY_FUNCTION** is only **used to verify password complexity** at the time of the user account creation or password change. It is **NOT USED for the user authentication** at the time of user logon to the database.

Working with Profiles

To modify an existing profile:

1. Start the **Profile Manager**.
2. Select the required profile in the profile list.
3. On the **General** tab page modify profile properties as needed
4. On the **Users** tab page modify user assignments as needed.



This is a multiple-choice screen. You can select as many users as you want. When you click the **Apply** button all of your changes including all changes on **General** page are applied at once.

5. To assign a user to the selected profile place a checkmark in the left-most column. To undo the assignment clear the corresponding checkmark.

 **Note:** User is automatically reverted to the DEFAULT profile if the previous profile assignment is cleared.

6. Click the **Apply** button to save changes.

To delete an existing profile:


1. Start the **Profile Manager**.
2. In the profile list select name of the profile you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the YES button if you are sure that the right profile is chosen for deletion.

 **Important Note:** **The DEFAULT profile may not be deleted!**

To create a new profile:

1. Start the **Profile Manager**.
2. Click the **New** button
3. On the **General** tab page modify profile properties as needed.
4. On the **Users** tab page assign users to the new profile as needed.
5. Click the **Apply** button to save changes.

Managing Space Usage Quotas

To view or modify user space quotas, click **Manage/Edit Space Usage Quotas** menu or click the Space Quota  button on the application toolbar. The **Tablespace Quotas** dialog will appear.

This dialog consists of two tab pages each providing a different view to the current database configuration:


- The first tab page **Space Quotas by User** displays space quotas grouped by user, in other words, for every user it shows all granted tablespaces and space quotas.
- The second tab page **Space Quotas by Tablespace** displays space quotas grouped by tablespace name, in other words, for every tablespace it shows all users who has been granted space in the tablespace space and their quotas.

You can modify settings using either By User or By Tablespace view, whichever is find more convenient to you. The end-results should be the same.


To modify an existing user space quota:

1. Activate **Space Quotas by User** tab page.
2. Click on the user whose tablespace quotas you want to modify in the left hand side tree-view
3. On the right hand side of the screen in the table-like grid control modify existing quotas.

or

To add another tablespace click the Insert  button on any existing line. A new row will appear above the clicked button. In the Tablespace Name column select required tablespace name in the drop-down list. Enter the required quota in the Space Quota column. The quota must be either a valid number or one of the following values: [DEFAULT], [UNLIMITED].

or


To completely revoke an existing tablespace quota use the Delete  button on the tablespace line whose quota you want to revoke from the user.

The top screenshot shows the 'Tablespace Quotas' dialog box with the 'Space Quotas By Tablespace' tab selected. The table below shows the current quotas for three tablespaces:

Tablespace Name	Space Type	Quota (MBytes)	Used (MBytes)
SYSAUX	NORMAL	100	
TEMP	TEMPORARY	[UNLIMITED]	
USERS	DEFAULT	[UNLIMITED]	
Total:		[UNLIMITED]	0


The bottom screenshot shows the 'Tablespace Quotas' dialog box with the 'Space Quotas By User' tab selected. The table below shows the current quotas for various users:

User Name	Space Type	Quota (MBytes)	Used (MBytes)
ANONYMOUS	DEFAULT	[UNLIMITED]	
BI	NORMAL	100	
CTXSYS	DEFAULT	[UNLIMITED]	5
DBSNMP	DEFAULT	[UNLIMITED]	0
DMSYS	DEFAULT	200	5
EXFSYS	DEFAULT	[UNLIMITED]	1
MDSYS	DEFAULT	[UNLIMITED]	6
MGMT_VIEW	DEFAULT	[UNLIMITED]	
OLAPSYS	DEFAULT	[UNLIMITED]	15
ORDPLUGINS	DEFAULT	[UNLIMITED]	
ORDSYS	DEFAULT	[UNLIMITED]	1
SI_INFORMTN_SCHEMA	DEFAULT	[UNLIMITED]	
SYSMAN	DEFAULT	[UNLIMITED]	48
WKPROXY	DEFAULT	[UNLIMITED]	
WKSYS	DEFAULT	[UNLIMITED]	7
WK_TEST	DEFAULT	[UNLIMITED]	12
WMSYS	DEFAULT	[UNLIMITED]	7
XDB	DEFAULT	[UNLIMITED]	47
Total:		[UNLIMITED]	154

 **Tip:** The UNLIMITED quota means to Oracle that the user can allocate as much space as much free space is available in the tablespace. If the tablespace files have AUTOEXTENT option turned on and the user requests more space than available Oracle automatically increases size of tablespace files to accommodate the new space requirements. The DEFAULT quota means to Oracle that the database server can decide how much space it can make available to the user.

4. Click the **Apply** button to save all changes.

Managing Resource Usage Quotas

To view or modify user resource quotas click **Manage/Edit Resource Usage Quotas** menu or click the Resource Quota  button on the application toolbar. The **Resource Quotas** dialog will appear.

User Name	Profile	Connect Time (min)	Idle Time (min)
ANONYMOUS	ANONYMOUS_001	600	30
BI	DEFAULT	UNLIMITED	UNLIMITED
CTXSYS	DEFAULT	UNLIMITED	UNLIMITED
DBSNMP	MONITORING_PROFILEDEFAULT	UNLIMITED	UNLIMITED
DB_AUDIT	DEFAULT	UNLIMITED	UNLIMITED
DIP	DEFAULT	UNLIMITED	UNLIMITED
DMSYS	DEFAULT	UNLIMITED	UNLIMITED
EXFSYS	DEFAULT	UNLIMITED	UNLIMITED
HR	DEFAULT	UNLIMITED	UNLIMITED
IX	DEFAULT	UNLIMITED	UNLIMITED
MDDATA	DEFAULT	UNLIMITED	UNLIMITED
MDSYS	DEFAULT	UNLIMITED	UNLIMITED

Connection Time Limits vs. Actual Usage

Minutes

Legend: Average (Red), Limit (Green), Maximum (Blue)

Maximum Connect Time: 600 (minutes)
Maximum Idle Time: 30 (minutes)

Note: If a user exceeds the Connect Time or Idle Time session resource limit, Oracle rolls back the current transaction and ends the session. When the user process next issues a call to Oracle, an error message is returned. Long-running queries and other operations are not subject to the "Idle Time" limit.

Tip: "Update Profile" action affects all users assigned to the selected profile. "Personalize" action saves the selected profile with a new name and then assigns the selected user to the new profile.

Buttons: Refresh, Personalize, Update Profile, Close

This dialog consists of several parts:

- The left top part contains table-like grid control listing all database users, their database profiles and profile resource usage limits.
- The left bottom part shows resource usage comparison chart for all database users. The chart changes according to the tab page selection on the right hand side of the dialog. In case if the **Connect & Idle Time** tab is selected then the chart shows comparison between actual user session timing statistics and configured user session time limits; in case if the **Input/Output (I/O) Limits** tab is selected then the chart changes to show comparison between actual user I/O statistics and configured I/O limits; and so on.

Tip: Use Actual vs. Limits comparison charts to make intelligent decisions on which resource usage limits needs to be set for different database users and what would be their optimal values.

Important Note: The "actual usage" statistics are collected by the background statistics collection package APP_MONITOR, which is part of [DB Application Monitor](#). You will not be able to use the resource usage comparison charts if the mentioned package is not installed in the database or the background statistics collection job is not running.

- The right hand side of the dialog contains 6 different tabs that can be used to setup different types of resource limits. These 6 tabs are described in detail in following paragraphs.

You can use standard DB Tools list sorting options to rearrange how user names or profile properties appear in the list. For more information about supported sorting options read the [Sorting](#) topic.

Working with Resource Quotas and Profiles

The left top part of the **Resource Quotas** dialog lists all database users, their database profiles and profile resource usage limits. Changes you make on the **Resource Quotas** dialog apply to the selected profile and not directly to the selected user. Because multiple users can be assigned the same database profile your changes made for a single profile can indirectly affect multiple users. To make it convenient for you to setup different resource usage limits for different users, without first creating a bunch of different profiles and assigning them to database users, the **Resource Quotas** dialog exposes 2 different methods for applying profile changes. Use the **Update Profile** button to save changes in the selected profile affecting all users assigned to that profile. Use the **Personalize** button to save changes as a brand new profile and automatically assign this new profile to the selected user only. In the last case DB Consumer Express automatically generates unique profile name in the *USERNAME_nnn* format, where *USERNAME* prefix is replaced with the selected user name and *nnn* suffix is replaced with a unique number.

You can use the [Profile Manager](#) tool to delete outdated profiles and change user/profile assignment

Resource Usage Limits

You can use 3 types of values for resource usage limits:

1. **UNLIMITED** - When specified with a resource parameter, UNLIMITED indicates that a user assigned the selected profile can use an unlimited amount of this resource.
2. **DEFAULT** - Specify DEFAULT if you want to omit a limit for this resource in this profile. A user assigned this profile is subject to the limit for this resource specified in the DEFAULT profile. The DEFAULT profile initially defines unlimited resources but you can change them if required.
3. **A numeric value** – Specify a number to indicate the maximum amount of this resource allowed for users assigned the selected profile.

Connect & Idle Time Limits

Maximum Connect Time - Specify the total elapsed time limit for a session, expressed in minutes.

Maximum Idle Time - Specify the permitted periods of continuous inactive time during a session, expressed in minutes.



Important Note: Long-running queries and other operations are not subject to this limit.

CPU Usage Limits

Maximum CPU Time (Per Session) - Specify the CPU time limit for a session, expressed in seconds.


Maximum CPU Time (Per Call) - Specify the CPU time limit for a call (a parse, execute, or fetch operation), expressed in seconds.

Maximum Concurrent Sessions - Specify the number of concurrent sessions to which you want to limit the users assigned the selected profile

Input/Output Limits


Maximum I/O (Per Session) - Specify the permitted number of Mbytes read in a session, including data read from memory and disk.

Maximum I/O (Per Call) - Specify the permitted number of Mbytes read for a call to process a SQL statement (a parse, execute, or fetch).

 **Tip:** Oracle database internally stores values of I/O limits expressed in the number of database blocks. DB Consumer Expert uses your database default block size to convert I/O limits from the number of data blocks to Mbytes.

Memory Usage Limits

Maximum Session Private Memory in SGA - Specify the amount of private space a session can allocate in the shared pool of the system global area (SGA), expressed in Kbytes.

 **Tip:** This limit applies only if you are using shared server architecture. The private space for a session in the SGA includes private SQL and PL/SQL areas, but not shared SQL and PL/SQL areas.

Composite Limits

Error! Bookmark not defined. Error! Bookmark not defined. Error!
Bookmark not defined.

Total resource cost - Specify the total resource cost for a session, expressed in service units. Oracle calculates the total service units as a weighted sum of CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION, and PRIVATE_SGA.

Important Notes:


- Initially weights for all resources are set to 0. Until you modify the initial setup the Composite Limits will not work. To modify resource weights click **Modify Unit Weights** button that will open up the **Composite Cost Limit Formula** dialog.

Composite Cost Limit Formula ✕

Specify cost parameters (relative weight) for a formula to calculate the total resource cost used in a session.

$TOTAL = (CPU_COST * CPU_TIME) + (CON_COST * CON_TIME) + (MBM_COST * MBM_AMOUNT) + (IO_COST * IO_AMOUNT)$

Resource Name	Resource Cost
Connect Time	<input style="width: 50px;" type="text" value="0"/>
CPU Time (per Session)	<input style="width: 50px;" type="text" value="0"/>
I/O (per Session)	<input style="width: 50px;" type="text" value="0"/>
Private Memory in SGA	<input style="width: 50px;" type="text" value="0"/>

 **Note:** Oracle calculates the total resource cost by multiplying the amount of each resource used in the session by the resource's weight and summing the products for all four resources. Both the products and the total cost are expressed in units called service units. The weight that you assign to each resource determines how much the use of that resource contributes to the total resource cost. If a session's cost exceeds the limit, Oracle aborts the session and returns an error.

Use this dialog to specify new resource weights.

- To exclude any particular resource type from the composite formula, enter **0** for that type. For example, if you assign a weight of 0 to the **Connect Time** resource, this resource no longer will affect the composite limit.
- To disable the composite formula completely, enter zeroes for all resource types.
- Weights of units entered in the **Composite Cost Limit Formula** apply to all database profiles having composite resource limit set to a value other than UNLIMITED. Oracle currently does not support using different formulas with different profiles.

Resource Usage Plans

Default Consumer Group - Specify the default consumer group you want to assign to the selected profile. All users assigned to this profile who have not been explicitly assigned a default consumer group will be affected by this change.



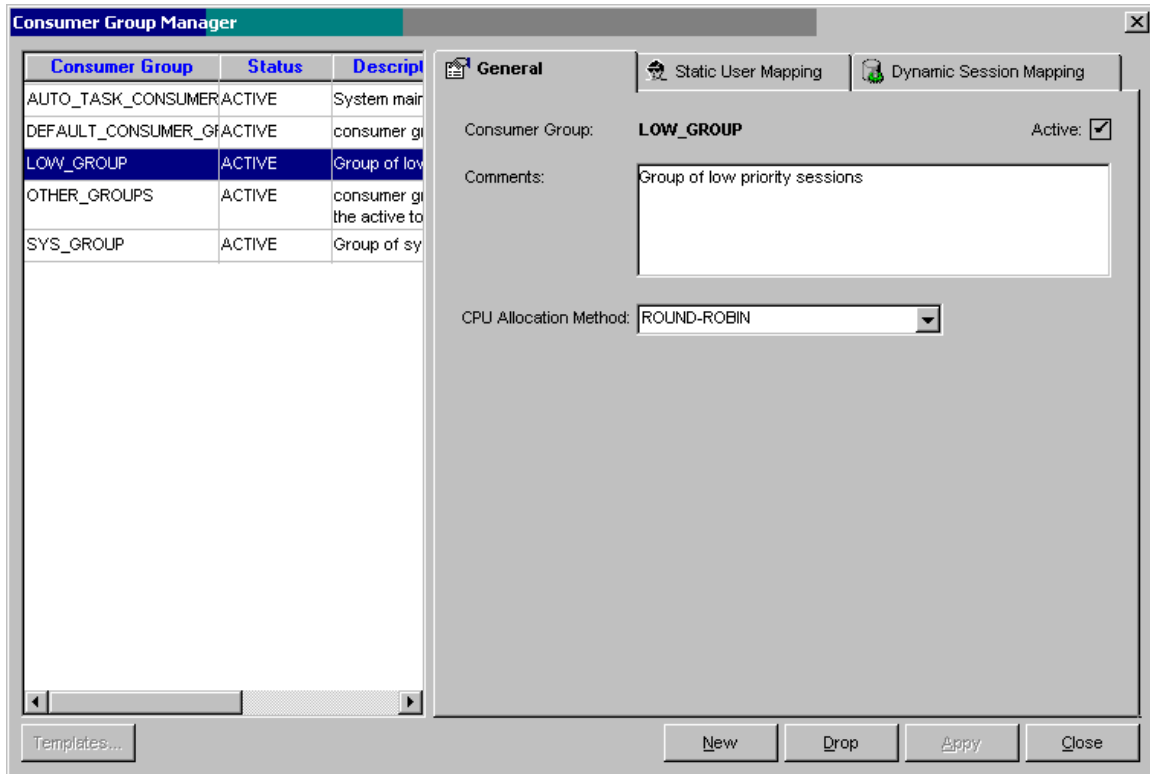
Tips:

- Resource usage limits set in profile properties affect all users assigned to the selected profile individually. Resource usage limits set for a consumer group affect all users as a group. That's it. All active users of a consumer group are subject to the collective limits set for that group.
- Use [Consumer Group Manager](#) and [Resource Plan Manager](#) tools to manage consumer groups and resource usage plans in the database.

Managing Consumer Groups

The Consumer Group Manager is only supported with Oracle 8i and up. Also note the Consumer Group Manager supports some options for Oracle 10g that are not available for Oracle 8i and 9i.

To start the Consumer Group Manager click **Manage/Consumer Groups** menu or click the Groups  button on the application toolbar. The **Consumer Group Manager** dialog will appear.



The Consumer Group Manager dialog consists of 2 parts: The left hand side is used to list all consumer groups and their properties in a single table-like list. The right hand side is used to display and edit group properties and user mapping settings. The settings and properties displayed on the right hand side are associated with the consumer group whose name is selected in the list on the left hand side.

You can use standard DB Tools list sorting options to rearrange how consumer group names or properties appear in the list. For more information about supported sorting options read the [Sorting](#) topic.

Consumer Group Properties

Consumer Group – Name of the consumer group

Active – Group status, either ACTIVE or PENDING. Normally the status should display ACTIVE all the time. You can encounter another status when somebody is trying to manipulate resource groups or plans manually by executing individual SQL commands in the database and the current setup is incomplete.

Comments – Description of the consumer group.

CPU Allocation Method– The allocation method for distributing CPU among sessions in the consumer group:

- The default is **ROUND_ROBIN** method, which uses a round-robin scheduler to ensure that sessions are fairly executed.
- **RUN_TO_COMPLETION** method specifies that sessions with the largest active time are scheduled ahead of other sessions. This method is not available in Oracle 8i and 9i.

Assigning Users and Sessions to Consumer Groups

Resource consumer groups are groups of users, or sessions, that are grouped together based on their processing needs. Users can be assigned to more than one resource consumer group, but each user's

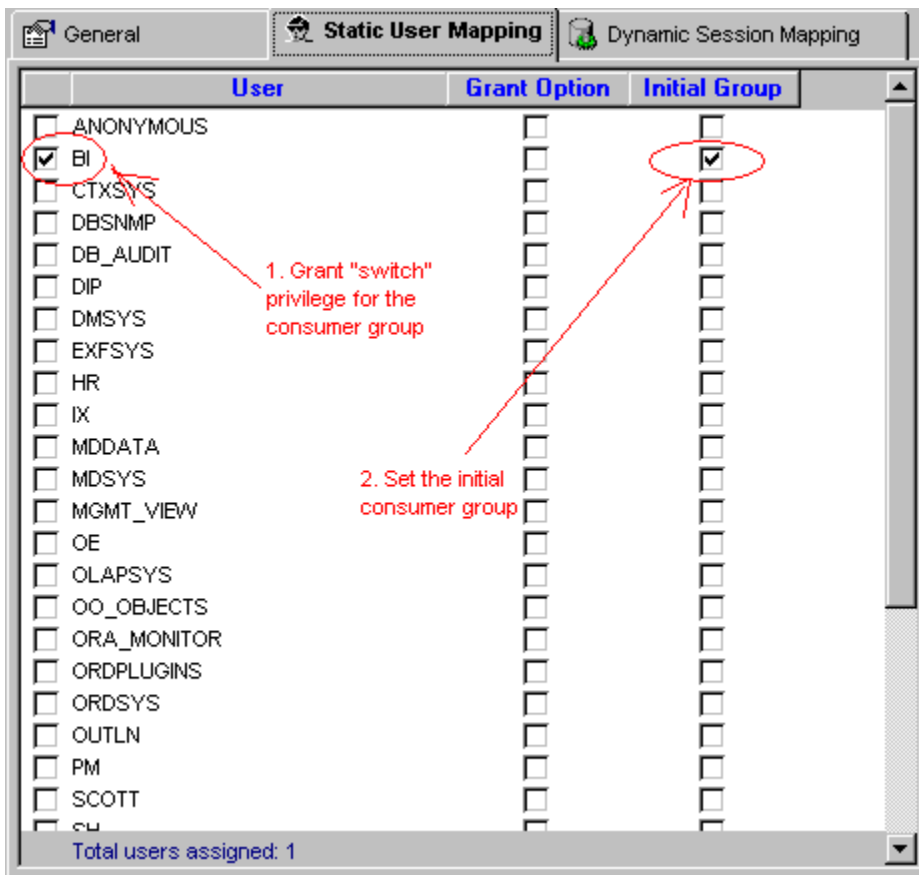
session can only be assigned to one resource consumer group at a time. For example, different consumer groups can be setup for OLTP users, for DSS users and for ad hoc queries. The same user can switch to different groups when running different database applications.

By default all users are assigned to the DEFAULT_CONSUMER_GROUP consumer group. Based on your requirements you can statically assign your users to specific groups using user name as a mapping method. For more information on static assignment methods see the following [Static Assignment](#) topic. You can also dynamically assign user sessions to different consumer groups based on some other parameters, for example, using application name or network user name as a mapping method. This way you can have the same user JOHN dynamically assigned to consumer group OLTP_USERS when John runs his daily transaction aggregation procedures and have him dynamically assigned to consumer group DECISION_SUPPORT when John runs his end-of-day summary reports. For more information on dynamic assignment methods see the following [Dynamic Assignment](#) topic.

Static Assignments

To have the same consumer group assigned to all database sessions for the same user, use the static assignment method. This method requires 2 simple steps: granting to the user "switch" privilege for the consumer group and then setting that consumer group as the user's initial consumer group. Here is how you can do that using the **Consumer Group Manager** tool.

1. On the **Consumer Group Manager** screen activate **Static User Mapping** tab page.




This is a multiple-choice screen. You can select as many users as you want. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

2. Grant users the privilege to use the selected consumer group. To grant user such privilege, place a checkmark in the left-most column. To revoke that privilege, clear the corresponding checkmark.

Checking the **Grant Option** box enables the selected user to grant this consumer group to someone else.


3. Check the **Initial Group** box to statically assign a user to the selected group.

 **Tip:** This option instructs Oracle to automatically treat all sessions for selected user as members of the selected consumer group.

Note: The **total** field shows how many users are checked on the screen.

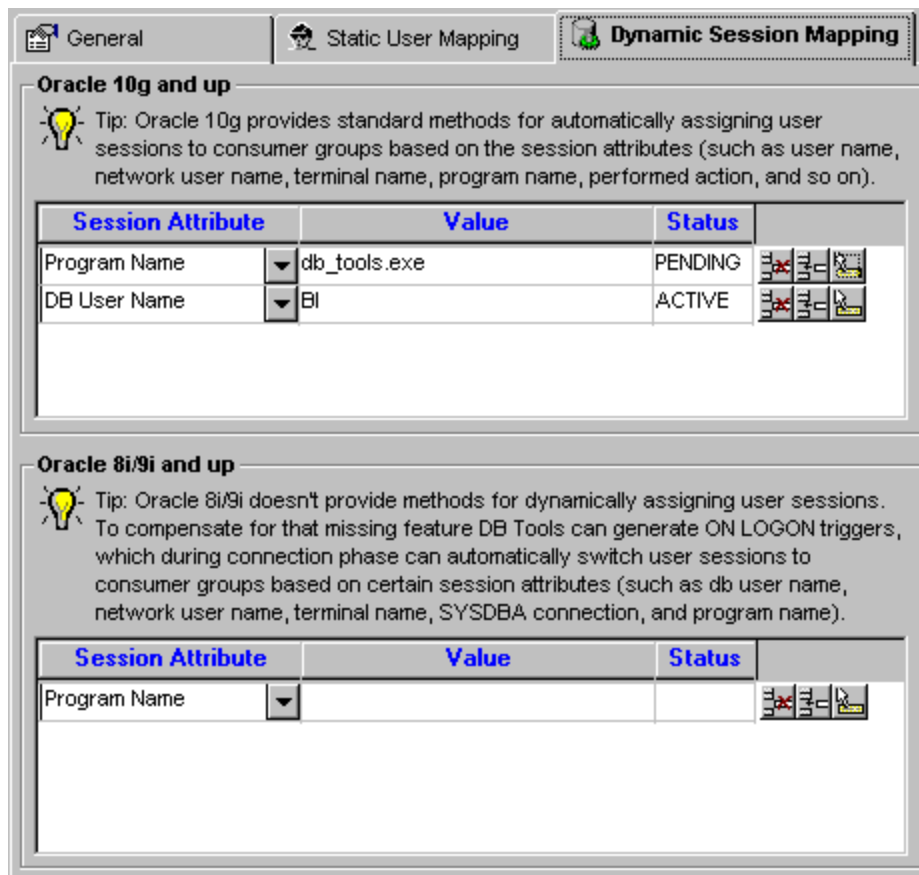
Dynamic Assignments

Resource consumer groups are groups of users, or sessions, that are grouped together based on their processing needs. Because some users can run multiple database applications with different processing needs you may want in different situations assign different groups to same user based on the name of the running application or some other parameters. This is called dynamic group assignment. The Database Resource Manager in Oracle 10g and up provides built-in native functions for dynamic session assignment. In Oracle 8i and 9i such functions don't exist that's why for Oracle 8i and 9i DB Consumer Expert provides an alternative solution using ON LOGON database triggers to dynamically switching consumer groups for new database sessions.


 **Tip:** This type of consumer group switching is completely dynamic and pertains to current sessions only; it is not persistent and also does not change the initial consumer groups for users.

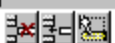
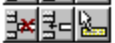
Here is how you can use the **Consumer Group Manager** tool to configure both methods:

1. On the **Consumer Group Manager** screen activate **Dynamic User Mapping** tab page.




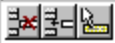
Oracle 10g and up

 Tip: Oracle 10g provides standard methods for automatically assigning user sessions to consumer groups based on the session attributes (such as user name, network user name, terminal name, program name, performed action, and so on).

Session Attribute	Value	Status	
Program Name	db_tools.exe	PENDING	
DB User Name	BI	ACTIVE	



Oracle 8i/9i and up


 Tip: Oracle 8i/9i doesn't provide methods for dynamically assigning user sessions. To compensate for that missing feature DB Tools can generate ON LOGON triggers, which during connection phase can automatically switch user sessions to consumer groups based on certain session attributes (such as db user name, network user name, terminal name, SYSDBA connection, and program name).


Session Attribute	Value	Status	
Program Name			


If you are running Oracle 10g or later use the top part of the screen labeled **Oracle 10g and up**. Although you can use the 8i/9i compatible method with Oracle 10g databases you should probably stick with the native method, as there are no advantages in using the other.

If you are running Oracle 8i or 9i use the bottom part of the screen labeled **Oracle 8i/9i and up**

2. Click the Insert  button to add a new line to the mapping table.
3. In the **Session Attribute** column select the type of session attribute you want to use for matching database sessions to the selected consumer group.
4. In the Value column type the attribute value or alternatively click the Lookup  button to browse currently active database sessions and pick up a value from the Lookup list.

 **Tip:** DB Consumer Expert displays different lookup lists for different attribute types. If, for example, you choose Program Name for the attribute type, DB Consumer Expert will display a list of all programs currently connected to the database. To make sure the program you are looking for does appear in the Lookup list ask somebody to run it for you while you are setting up the dynamic session assignment.

5. Repeat steps 2 to 4 for other attributes or values. Use the Delete  button to delete unnecessary assignments.

 **Important Note:** It is highly recommended that in Oracle 8i and 9i you grant the "switch" privilege for all "dynamic" consumer groups to PUBLIC in order to avoid a situation when a new session gets switched to a consumer group for which the session user doesn't have the switch privilege. Such session would then miss the required consumer group.

Creating New Consumer Groups

1. Open **Consumer Group Manager** dialog using **Manage/Manage Consumer Groups** menu.
2. Click the **New** button.
3. Fill in new group properties on the **General** tab page.
4. Enter additional options and setup user/group assignments using other tab pages. For detailed information on how to use other tab pages read previous topics in this chapter.
5. Click the **Apply** button to save changes.

Deleting Consumer Groups

1. Open **Consumer Group Manager** dialog using **Manage/Manage Consumer Groups** menu.
2. On the left hand side of the dialog select name of the consumer group you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the YES button if you are sure that the right group is chosen for deletion.

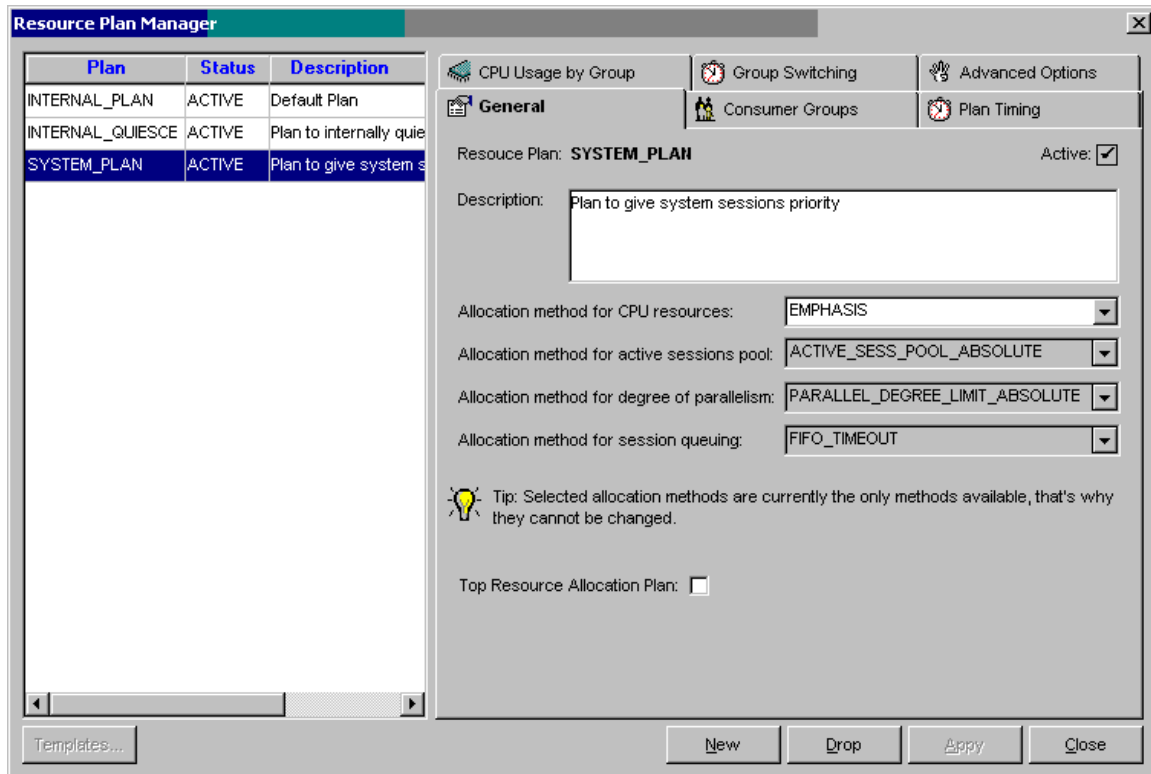
Important Notes:

- Any existing user that has the deleted group set as the default group will be automatically reassigned to the DEFAULT_CONSUMER_GROUP.
- Certain consumer groups cannot be deleted. They include DEFAULT_CONSUMER_GROUP, SYS_GROUP and OTHER_GROUPS.

Managing Resource Usage Plans

The Resource Plan Manager is only supported with Oracle 8i and up. Also note the Resource Usage Plan Manager supports some options for Oracle 10g that are not available for Oracle 8i and 9i.

To start the Resource Usage Plan Manager click **Manage/Resource Plans** menu or click the Plans  button on the application toolbar. The **Resource Plan Manager** dialog will appear.



The Resource Plan Manager dialog consists of 2 parts: The left hand side is used to list all resource usage plans and their properties in a single table-like list. The right hand side is used to display and edit plan properties and consumer group assignments. The settings and properties displayed on the right hand side are associated with the resource usage plan whose name is selected in the list on the left hand side.

You can use standard DB Tools list sorting options to rearrange how consumer group names or properties appear in the list. For more information about supported sorting options read the [Sorting](#) topic.


Resource Plan Properties

Resource Plan – Name of the resource usage plan.

Active – Plan status, either ACTIVE or PENDING. Normally the status should display ACTIVE all the time. You can encounter another status when somebody is trying to manipulate resource groups or plans manually by executing individual SQL commands in the database and the current setup is incomplete.

Comments – Description of the resource plan.

CPU Allocation Method – The allocation method for distributing CPU among consumer groups.

 **Tip:** Do not confuse this property with the CPU allocation method within consumer groups. CPU allocation method within consumer group controls in which order database sessions get CPU time, while CPU allocation method for the plan controls how much of a share of total CPU time is given to different consumer groups. These shares are often called "service levels." Eight (8) levels are supported by the Oracle Database Resource Manager and their numbering ranges from 1 (highest level) to 8 (lowest level). Consumer groups assigned to levels with higher numbers get more CPU time than groups assigned to lower levels. Multiple consumer groups can be assigned to the same or different level as well as one consumer group can be assigned to multiple levels in the same plan.

For example, if both consumer groups A and B are assigned to level 2 they will get equal share of CPU time, however if group A consists of 2 database sessions and group B consists of 10 database sessions and both groups use the same ROUND-ROBIN method for CPU allocation between sessions then obviously each session in group A will get 5 times more CPU than any session in group B


- The default is **EMPHASIS** method, which should be used for multilevel plans that use percentages to specify how CPU is distributed among consumer groups.
- **RATIO** is for single-level plans that use ratios to specify how CPU is distributed. This method is not available in Oracle 8i and 9i.

Active Session Pool Resource Allocation Method – Limits the number of active sessions. All other sessions are inactive and wait in a queue to be activated. `ACTIVE_SESS_POOL_ABSOLUTE` is the default and only method available in Oracle 8i, 9i and 10g.

Degree of Parallelism Allocation Method – Limits the degree of parallelism of any operation. `PARALLEL_DEGREE_LIMIT_ABSOLUTE` is the default and only method available in Oracle 8i, 9i and 10g.

Session Queuing Allocation Method – Controls order in which queued inactive sessions will execute. `FIFO_TIMEOUT` is the default and only method available.

Top Resource Allocation Plan – Controls which plan is the top plan. Only one top plan exists at any moment in time. If no top plan is selected the Oracle Database Resource Manager effectively becomes disabled.

 **Tip:** The **Top Resource Allocation Plan** property applies to the Database Resource Manager as a whole rather than to the selected resource plan. This option is displayed together with the resource plan properties for your convenience only.

Linking Consumer Groups to Resource Plans


Before you can assign consumer groups to different services levels or specify any resource usage limits you must specify which consumer groups will participate in the resource usage plan. Use the **Consumer Groups** tab page available on the **Resource Plan Manager** dialog to link groups and plans.

CPU Usage by Group		Group Switching		Advanced Options	
General		Consumer Groups		Plan Timing	
Consumer Group	CPU Method	Status	Description		
<input type="checkbox"/>	AUTO_TASK_CONSUMER_ROUND-ROBIN	ACTIVE	System maintenance task cor		
<input type="checkbox"/>	DEFAULT_CONSUMER_GR.ROUND-ROBIN	ACTIVE	consumer group for users nc group		
<input checked="" type="checkbox"/>	LOW_GROUP	ROUND-ROBIN	ACTIVE	Group of low priority session	
<input checked="" type="checkbox"/>	OTHER_GROUPS	ROUND-ROBIN	ACTIVE	consumer group for users nc group in the active top-plan	
<input checked="" type="checkbox"/>	SYS_GROUP	ROUND-ROBIN	ACTIVE	Group of system sessions	

Total groups assigned: 3

This is a multiple-choice screen. You can select as many groups as you want. When you click the Apply button all of your changes including changes on other pages are applied at once.

To link a group to the selected resource plan, place a checkmark in the left-most column. To unlink that group, clear the corresponding checkmark.

 **Important Notes:** The **OTHER_GROUPS** consumer group must be linked to every resource plan. This ensures that a session which is not part of any of the consumer groups included in the currently active plan is allocated resources (as specified by the OTHER_GROUPS).

Note: The total field shows how many groups are checked on the screen.

Assigning CPU Service Levels to Consumer Groups

Use the **CPU Usage by Group** tab on the **Resource Plan Manager** dialog to allocate CPU time to linked consumer groups. Note that only groups linked to the selected plan appear in the **CPU Service Levels** table. To add or remove groups use the **Consumer Groups** tab page whose usage is described in the previous topic.

General Consumer Groups Plan Timing
 CPU Usage by Group Group Switching Advanced Options

Note: For "EMPHASIS" CPU allocation method you can use all supported CPU service levels from 1 to 8. The service level represents % of CPU time and priority allocated for a consumer group.
 For "RATIO" CPU allocation method (available in Oracle 10g and up) you can use only the first CPU service level which specifies the weight of CPU usage.

Tips: CPU allocation method can be selected on the General tab page.
 For multilevel plans you should always select "EMPHASIS" CPU allocation method in order to use percentages to specify how CPU is distributed among consumer groups. You may select "RATIO" method for single-level plans if you want to use ratios to specify how CPU is distributed.

CPU Service Levels (% of CPU Time Allocation)

Consumer Group	Level 1	Level 2	Level 3	Level 4	Level 5
SYS_GROUP	100%				
OTHER_GROUPS		100%			
LOW_GROUP			100%		
Total:	100%	100%	100%	0%	0%

For every listed consumer group enter percent of CPU time allocated in the group in one or more service levels. The number must be between 0 and 100 but the horizontal sum for every group must be a positive number. Note that each group can be assigned to one or more levels. In case if you are going to assign all groups to the same level, in other words, use single-level plan, you can then use RATIO for the CPU allocation method on the General page.

The totals on the bottom of the CPU Service Levels table show total CPU allocation for each level. This number may not exceed 100%.

After making CPU allocation changes do not forget to click the **Apply** button to save your changes including changes on other pages, which are all applied at once.

Example of a single-level CPU allocation plan

CPU Service Levels (% of CPU Time Allocation)

Consumer Group	Level 1	Level 2	Level 3	Level 4	Level 5
SYS_GROUP	50%				
OTHER_GROUPS	40%				
LOW_GROUP	10%				
Total:	100%	0%	0%	0%	0%

Example of a multi-level CPU allocation plan with shared levels


Consumer Group	Level 1	Level 2	Level 3	Level 4	Level 5
SYS_GROUP	50%	25%	25%		
OTHER_GROUPS		40%	60%		
LOW_GROUP				100%	
Total:	50%	65%	85%	100%	0%

Limiting Resources Available to Consumer Groups

Use the **Advanced Options** tab on the **Resource Plan Manager** dialog if you want to setup advanced resource allocation limits for each consumer group linked to the selected resource plan. Note that only groups linked to the selected plan appear in the **Advanced Options** table. To add or remove groups use the **Consumer Groups** tab page whose usage is described in the previous topic.

General
Consumer Groups
Plan Timing

CPU Usage by Group
Group Switching
Advanced Options

 **Tips:**

(1) Parallel Degree Limit -- Specifies a limit on the degree of parallelism for any operation.

(2) Active Sessions Pool Size -- Specifies a limit on the number of concurrently active sessions for a consumer group. When the limit is reached, new operations are queued and wait for completion of the active operations.

(3) Undo Pool Size (Oracle9i and up) -- Specifies a limit on the maximum size of undo space allocated by a consumer group. When the limit is reached, operations are suspended and queued until the space become available.

(4) Queue Time Limit (Oracle 9i and up) - Specifies time limit after which a job in an inactive session queue (waiting for execution) will time out.

(5) Maximum Idle Time Limit (Oracle 10g and up) -- Specifies a limit on the maximum session idle time. When the limit is reached, the session is automatically terminated.


(6) Maximum Blocker Session Idle Time Limit (Oracle 10g and up) -- Specifies a limit on session the maximum session idle time of a blocking session. When the limit is reached, the session is automatically terminated and locks are released.

Consumer Group	Parallel Degree Limit	Active Sessions Pool Size	Waiting Session Queue Timeout (seconds)	Maximum Pool Size (K)
SYS_GROUP	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED
OTHER_GROUPS	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED
LOW_GROUP	UNLIMITED	UNLIMITED	UNLIMITED	UNLIMITED

The complete description of each available parameter is provided directly on the **Advanced Options** tab page above the "resource usage limits by group" table.


After setting new resource limits do not forget to click the **Apply** button to save your changes including changes on other pages, which are all applied at once.

Dynamic Switching Between Consumer Groups

 **Note:** Options described in this topic are not available in Oracle 8i and prior.

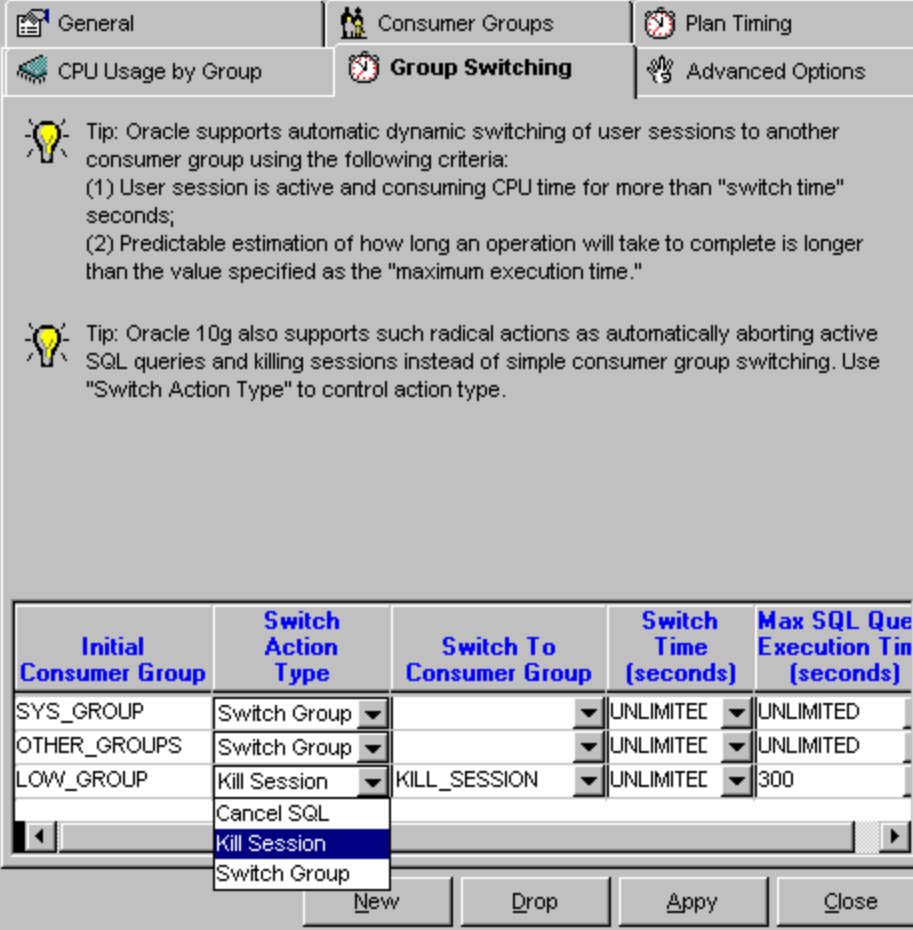
Database sessions can be dynamically switched between different resource consumer groups in the event they exceed certain resource limits. For example, instead of killing a session of a user who is using excessive CPU, the Database Resource Manager can instead change that user's consumer group to one that is allowed less CPU. This kind of resource-usage based consumer group switching can be enforced automatically by the Oracle Database Resource Manager.

Do not confuse this type of session switching with the initial [dynamic session assignment](#) to different consumer groups which is based on the session connection properties such as database program name, network user name, module name, and so on.

 **Tip:** The resource-usage based consumer group switching is completely dynamic and pertains to current sessions only; it is not persistent and also does not change the initial consumer groups for users.

Use the **Group Switching** tab on the **Resource Plan Manager** dialog to setup dynamic upgrades and downgrades of running user sessions between consumer groups and in the end between service levels.

Note that only groups linked to the selected plan appear in the **Advanced Options** table. To add or remove groups use the **Consumer Groups** tab page whose usage is described in the previous topic.



Initial Consumer Group	Switch Action Type	Switch To Consumer Group	Switch Time (seconds)	Max SQL Que Execution Tin (seconds)
SYS_GROUP	Switch Group		UNLIMITEC	UNLIMITED
OTHER_GROUPS	Switch Group		UNLIMITEC	UNLIMITED
LOW_GROUP	Kill Session	KILL_SESSION	UNLIMITEC	300

The complete description of each available parameter and option is provided directly on the **Group Switching** tab page above the "group switching rules" table.

After setting new consumer group switching rules do not forget to click the **Apply** button to save your changes including changes on other pages, which are all applied at once.



Note: **Cancel SQL** and **Kill Session** options are not available in Oracle 9i.

Time-based Resource Plan Activation

Only one resource usage plan can be active per Oracle database instance at any moment. Because not always "one size fits all" you may want to activate different plans in different scenarios. For example, you can define many different resource plans—one for daytime operations, one for nighttime operations, one for peak time processing and so on.

Use the **Plan Timing** tab on the **Resource Plan Manager** dialog to schedule resource plan automatic switching. Note that at least 2 different plans must be setup with different switch times in order for the plan switching to work correctly.

The screenshot shows the 'Plan Timing' tab of the Resource Plan Manager dialog. It includes a tip about automatic switching, a 'First Switch Date' field set to 01/24/2005, and a 'Switch Time' field set to 23:00:00. Under 'Switch Schedule', the 'Repeat daily' option is selected. Below this, there are checkboxes for days of the week (Sun, Mon, Tue, Wed, Thu, Fri, Sat) and other options like 'Repeat monthly', 'Run all day, repeat every' (set to 00:00:00), and 'Custom recursion interval' (set to trunc(SYSDATE + 1) + 1/86400*82800). At the bottom, there is a 'Switch Job Definition' text area containing a PL/SQL block and two buttons: 'Schedule Job' and 'Delete Job'.

The **Plan Timing** tab page provides limited interface to the [DB Job Expert](#), which allows you schedule resource usage plan switching with just a few mouse clicks.

Select appropriate schedule type and options and then click the **Schedule Job** button to schedule a new job.

To remove previously scheduled job, click the **Delete Job** button.



Notes:

- Deleting an existing resource plan switch job only removes that job definition from the job queue. It does not change the active resource plan and does not delete the affected plan from the database.
- Job scheduling and deletion have take immediate effect. You do not need to click the **Apply** button to apply the schedule. However, if you are scheduling a job for a new plan you must click the **Apply** button to save your changes.

Enabling/Disabling Database Resource Manager

Because only one resource usage plan can be active per Oracle database instance at any moment you can easily enable and disable the Database Resource Manager by selecting any existing plan as a top plan or selecting no plans at all. To select /deselect a resource plan use the **Top Resource Allocation Plan** checkbox available on the **General** page of the **Resource Plan Manager** dialog. See [Resource Plan Properties](#) topic for more information about **Top Resource Allocation Plan** option.



Important Note: Make sure to delete all [automated resource plan switch jobs](#) if your intent is to completely disable the Database Resource Manager.

Creating New Resource Plans

1. Open **Resource Plan Manager** dialog using **Manage/Manage Resource Plans** menu.
2. Click the **New** button.
3. Fill in new plan properties on the **General** tab page.
4. Activate **Consumer Groups** page and choose which groups can participate in the new plan.
5. Enter additional options and settings using other tab pages. For detailed information on how to use other tab pages read previous topics in this chapter.
6. Click the **Apply** button to save changes.

Deleting Resource Plans

1. Open **Resource Plan Manager** dialog using **Manage/Manage Resource Plans** menu.
2. On the left hand side of the dialog select name of the resource plan you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the YES button if you are sure that the right plan is chosen for deletion.



Important Notes: Certain resource plans cannot be deleted. They include SYSTEM_PLAN and plans whose name begins with "INTERNAL_" prefix..


Security Navigator

The **Security Navigator** is a comprehensive utility for exploring Oracle database security. Using that utility you can get a hierarchical display of security information presented in a convenient, easy understandable format. You can interactively browse grants made and received by all database users including cascading and recursive grants received from other users and roles.

Use the Security Navigator to discover security holes in your database.

The Security Navigator window is divided into two parts presented by 2 separate tab pages:


- The "Users, Grants, Objects" tab
- The "Roles, Grants" tab

To start the Security Navigator select **Manage/Security Navigator** menu or alternatively click the Security  button on the application toolbar.

Navigating

To navigate the Security Navigator use standard tree-view navigation techniques that you use when working with Windows Explorer and other applications. For quick expanding or collapsing of multiple items use right mouse button clicks to invoke the popup menu. Use the following commands available in the popup menu to quickly expand or collapse multiple items:

- **Expand Branch**
- **Collapse Branch**
- **Expand All**
- **Collapse All**

 **Note:** Avoid using **Expand All** command for the entire database if your database is large because adding all your objects and security settings to the Security Navigator tree-view could take a long time and the expanded tree could be slow to navigate.

Which items are displayed?

Security Navigator displays only context sensitive items. For example, Security Navigator will display **Grants Made** folder for the expanded user only if that user has granted something to other users or roles. When expanded the **Grants Made** folder will display all grants made by the mentioned user.

Which items are not displayed?

Security Navigator displays only context sensitive items. For example, Security Navigator will not display **Grants Made** folder for the expanded user if that user has granted nothing to other users or roles.

Zooming

You can use zoom feature (select **View/Zoom** command) to zoom in/out security information. Note that new zoom will be applied to the active tab.

Printing

You can print information displayed in the Security Navigator tabs by choosing File/Print command (shortcut: CTRL P). You can preview how it will look when printed by choosing File/Print Preview. Note that print and print preview commands will be applied to the active tab. See [Print](#) topic for information on print and print preview options.

Reports

DB Consumer Expert provides a large number of resource usage allocation and usage trends reports. Each report can be viewed and printed from the Report window. Many of these reports are based on the application performance statistics collected by the [Application Monitor package](#).

The following reports are available:

General

[Users](#)
[Profiles](#)

Space Usage by User

[Space Usage By User](#)
[User Exhausting Space Quota](#)

Resource Usage Summary by User

[Top 10 Users by Space Usage](#)
[Top 10 Users by CPU Usage](#)
[Top 10 Users by Memory Usage](#)
[Top 10 Users by Disk I/O](#)

Space and Resource Usage Trends by User

[Space Usage Trends](#)
[CPU Usage Trends](#)
[Memory Usage Trends](#)
[I/O Usage Trends](#)

Resource Usage Summary by Application

[Top 10 Applications by CPU Usage](#)
[Top 10 Applications by Memory Usage](#)
[Top 10 Applications by Sorts](#)
[Top 10 Applications by Disk I/O](#)

Resource Usage Trends by Application

[Application Resource Consumption](#)
[Application Performance & Trends](#)
[Database Load Balancing](#)

You can use them to analyze database resource usage by applications and users, spot application trends, tune and troubleshoot applications and consumer groups, tune database resource plans and improve overall database resource usage. The group of trend reports can be also used for gathering information required for proper database capacity planning.

Click on a report name in the list above for more information.

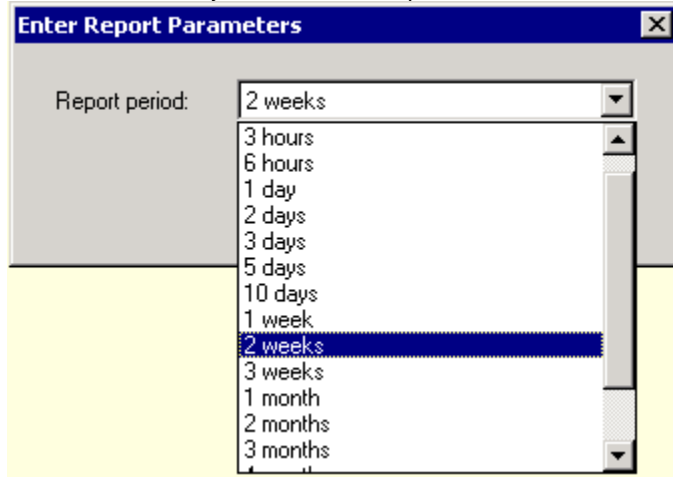
**Tips:**

- Most reports use application performance data for application processing analysis. This data is not available automatically. To allow application performance data collection you must install the back-end [Application Monitor](#) package and job. Do not run summary and trends reports immediately after you install these back-end components. Wait as long as necessary for each application to run at least 4-5 times. This will allow the Application Monitor to gather sufficient volume of application performance data for an accurate application analysis. Do not force/trigger applications manually for the purpose of performance data collection, as it will lead to inaccurate reports. The accuracy of reports is proportional to the volume of gathered data. Longer you wait, more applications run, and more valuable application performance data can be gathered.
- Application performance data is stored in the ORA_MONITOR.APP_STAT table. If for some reasons you cannot find a pre-built application report that fits your needs, feel free to create your own reports using the [Performance Library](#) tools. For description and usage of Application Monitor repository tables see source code for the Application Monitor package provided with DB Tools.

Viewing reports:

- 1 From the Report menu, choose the type of report you want. The report window will appear.

- 2 If the selected report analyses historical performance data, you will be prompted to choose time interval for the analysis. Select the required interval from the drop-down list and click OK.



- 3 You can use the Zoom feature to zoom in and out. You can either select one of the predefined values in the zoom drop-down box or type in the value you want then press the Enter key.

Printing reports:

- 1 Run report as described above.
- 2 When the report is ready, click the Print  toolbar button or select **File/Print** command from the main menu

Users Report

This reports displays detailed information about all database users and their properties.

Profiles Report

This report displays detailed information about all database profiles, including associated password security settings and resource usage limits.

Space Usage by User Report

This report shows summary usage statistics for each database user, including actual space usage, user space quotas by tablespace and percentage of that space used.

Users Exhausting Space Quotas Report

This report shows summary usage statistics for each database user who may soon run out of space. The report analyses users having 90% or more of their space used. It displays actual space usage by user, space quotas by tablespace and percentage of that space used.

Top 10 Users by Space Usage

This report displays summary pie chart for top-10 users by space allocated for various database objects.

You can use this report to identify top 10 users consuming most of the database space.

Top 10 Users by CPU Usage

This report provides several different views focusing on top 10 users whose sessions consumed most CPU time during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify top 10 CPU consuming users. This information can help you in setting up consumer groups and dynamic or static session mapping to consumer groups.

This report is similar to [Top 10 Applications by CPU Usage](#) report whose data is analyzed and grouped by application rather than by user name

See [Top 10 Applications by CPU Usage](#) topic in the DB Application Expert chapter for more information about this report and usage instructions.

Top 10 Users by Memory Usage

This report calculates memory usage chart focusing on top 10 users whose sessions consumed most SGA private memory during the chosen report period. It shows both maximum and average memory usage numbers. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify top 10 memory consuming users. This information can help you in setting up consumer groups and dynamic or static session mapping to consumer groups.

This report is similar to [Top 10 Applications by Memory Usage](#) report whose data is analyzed and grouped by application rather than by user name

See [Top 10 Applications by Memory Usage](#) topic in the DB Application Expert chapter for more information about this report and usage instructions.

Top 10 Users by Disk I/O (Input/Output)

This report calculates 2 charts focusing on top 10 users whose sessions caused most I/O during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify top 10 resource consuming users. This information can help you in setting up consumer groups and dynamic or static session mapping to consumer groups.

This report is similar to [Top 10 Applications by Disk I/O](#) report whose data is analyzed and grouped by application rather than by user name

See [Top 10 Applications by Disk I/O](#) topic in the DB Application Expert chapter for more information about this report and usage instructions.

Space Usage Trends

This report shows trend line chart for the database space allocation by all users during the selected report period. This report is based on statistical data collected by [DB Trends Expert](#).

You can use this report to identify database space usage trends and use this information for database capacity planning.

CPU Usage Trends

This report shows trend chart for the total system load created by all user sessions that ran during the selected report period and also shows the system load created by individual users. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify database CPU usage trends and use this information for database capacity planning.

Memory Usage Trends

This report shows trend chart for the total memory allocations for all user sessions that ran during the selected report period and also shows the summary memory requirements for individual users. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify database memory usage trends and use this information for database capacity planning.

I/O Usage Trends

This report shows trend chart for the total I/O generated by all user sessions that ran during the selected report period and also shows the summary I/O numbers generated by individual users. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify database I/O usage trends and use this information for database capacity planning.

Application Resource Consumption Report

This report shows trend chart for the total system load created by all applications that ran during the selected report period and also shows the system load created by individual applications. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report for the following purposes:

- 1 To identify applications consuming lots of system resources.
- 2 To evenly distribute your system load between different applications so that the resource intensive

applications do not run at the same time. This information can help you in improving the overall system performance, improving application response times and meeting required service levels.

See [Application Resource Consumption Report](#) topic in the DB Application Expert chapter for detailed description of this report and usage instructions.

Application Performance and Trends Reports

This report presents trend-line charts for top 20 applications whose average run-time duration have changed during the selected report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report for the following purposes:

- 1 Identifying applications whose run-time duration significantly changed from run-to-run during the chosen report period.
- 2 Identifying application duration trends (i.e., how quickly the duration changes over time) and thus you can use this information to forecast future durations.

See [Application Performance and Trends Report](#) topic in the DB Application Expert chapter for detailed description of this report and usage instructions.

Database Load Balancing Report

The Database Load Balancing Report can help you identify time windows of high and low server utilization and setup new or modify existing applications to run during low utilization times. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

The report contains two system load charts: the "load by day of week" chart and the "load by time of day" chart. Both charts display summary database utilization (% of CPU time) for all applications executed during the chosen report period.

See [Database Load Balancing Report](#) topic in the DB Application Expert chapter for detailed description of this report and usage instructions.

Top 10 Applications by CPU Usage

This report provides several different views focusing on top 10 applications that consumed most CPU time during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify top 10 CPU consuming applications. This information can help you in setting up consumer groups and dynamic or static session mapping to consumer groups.

See [Top 10 Applications by CPU Usage](#) topic in the DB Application Expert chapter for detailed description of this report and usage instructions.

Top 10 Applications by Memory Usage

This report calculates memory usage chart for applications that ran during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify top 10 memory consuming applications. This information can help you in setting up consumer groups and dynamic or static session mapping to consumer groups.

See [Top 10 Applications by Memory Usage](#) topic in the DB Application Expert chapter for detailed description of this report and usage instructions.

Top 10 Applications by Sorts

This report calculates 2 charts for applications that have run during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify top 10 resource consuming applications. This information can help you in setting up consumer groups and dynamic or static session mapping to consumer groups.

See [Top 10 Applications by Sorts](#) topic in the DB Application Expert chapter for detailed description of this report and usage instructions.

Top 10 Applications by Disk I/O (Input/Output)

This report calculates 2 charts for applications that have run during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report to identify top 10 resource consuming applications. This information can help you in setting up consumer groups and dynamic or static session mapping to consumer groups.

See [Top 10 Applications by Disk I/O](#) topic in the DB Application Expert chapter for detailed description of this report and usage instructions.

DB Security Expert

Overview

DB Security Expert provides innovative approach to analyzing and managing database security. Most native Oracle database utilities and commercial security solutions present database security settings simply as a bunch of users, roles and their privileged granted directly to them. In reality the database security is interdependent multi-layered system. Actual permissions and access level available to a database user may be inherited recursively from roles granted to roles, permissions granted to roles, permissions granted to access views, executed procedures which in turn allow users to access the data. The core of the DB Security Expert is Effective Security Settings Wizard allowing you to analyze these multi-layered effective security settings and assess the true state of data access privileges. In addition, the DB Security Expert provides tools for managing Oracle database users, roles, profiles and other security objects and settings in Oracle databases.

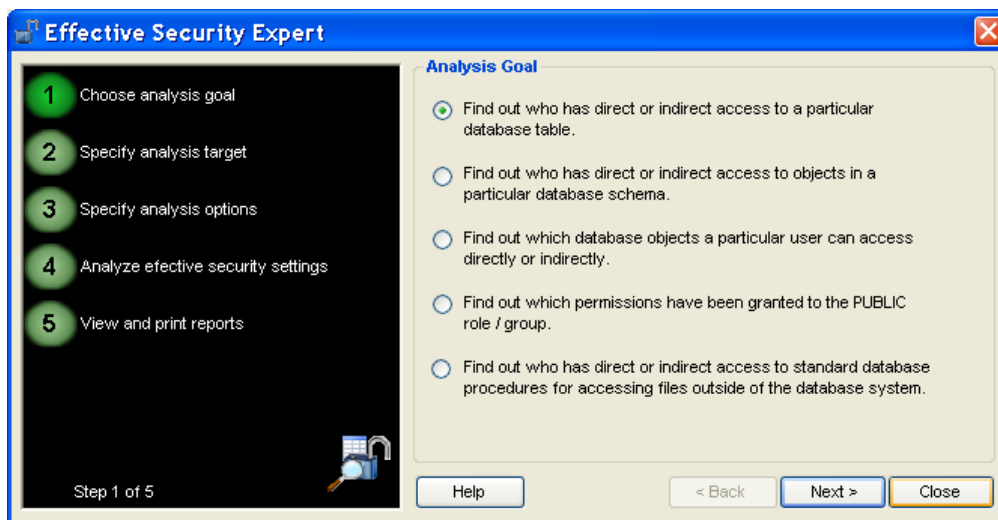
Analyzing Effective Security Settings

Using the Effective Security Settings utility, you can:

- Analyze effective access rights for any given user, table or an entire schema
- Find out how users gained rights to a particular database table or objects within a particular database schema. Whether they can access them directly or indirectly via assigned roles, group membership, even via database views
- Analyze effective access rights granted to the PUBLIC role.
- Analyze who has access rights to standard database procedures that can be used for accessing Operation System files outside of the database.
- Use the results to ensure enforcement of the security policies

To launch Effective Security Wizard:

1. Click Tools/Effective Security menu. The following dialog will appear




2. In the **Select Goal** step, select the goal for security settings analysis. Click the Next button to advance to the next step

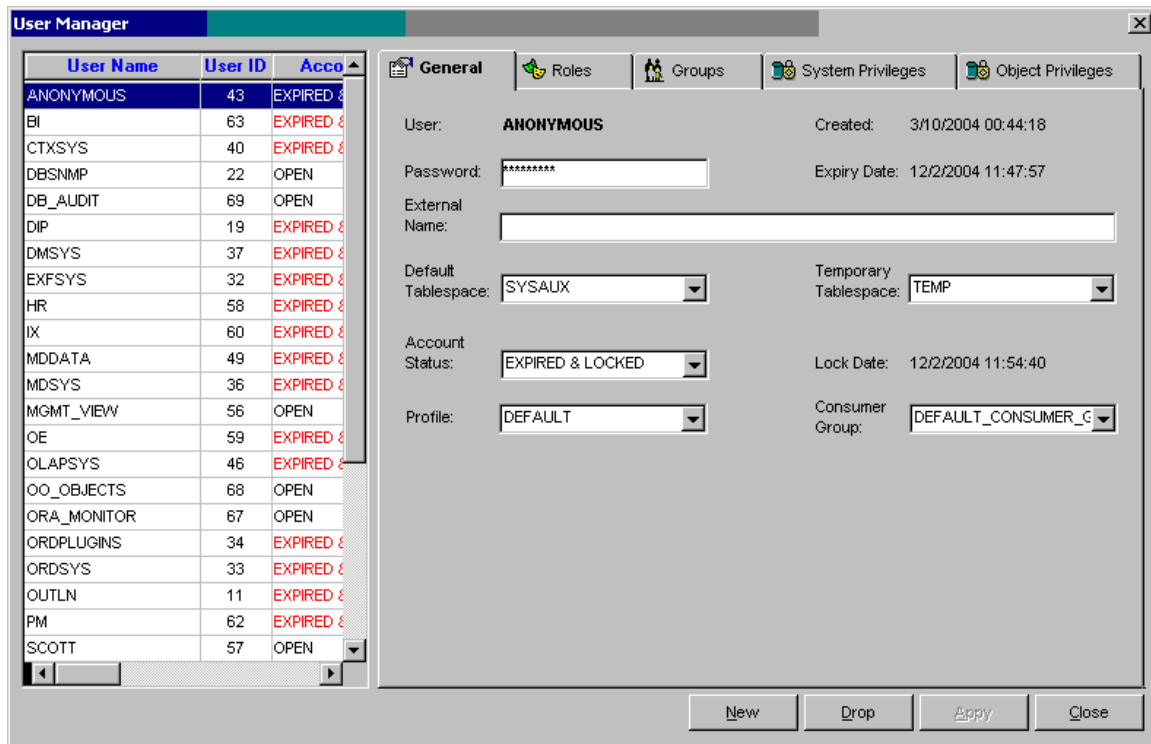
3. In the **Select Target** step, select the target database, schema, object or user whose security settings or access you want to check. Click the Next button to advance to the next step
4. In the **Select Analysis Options** step, select the required analysis options. Click the Next button to start the analysis.

To save analysis results to an external text file:

1. Click the File/Save menu or Save icon on the toolbar. The **Select Output File** dialog will appear.
2. Choose the output file name and location and click the OK button on the Select File dialog

Managing Database Users

To start the User Manager click **Manage/Manage Users** menu or click the Users  button on the application toolbar. The **User Manager** dialog will appear.



The User Manager dialog consists of 2 parts: The left hand side is used to list all database users and their properties in a single table-like list. The right hand side is used to display and edit user properties assign roles and assign various system-level and object-level privileges. The settings and properties displayed on the right hand side are associated with the user whose name is selected in the user list on the left hand side.

You can use standard DB Tools list sorting options to rearrange how user names or properties appear in the list. For more information about supported sorting options read the [Sorting](#) topic.

User Properties


Different Oracle database versions support different sets of user properties. On the **General** tab page DB Security Expert displays only options supported in the database you are connected to.

User – Specify Oracle user account name.


Lock Date – Date and time when the user account was created. This property is automatic and cannot be changed.

Password – Specify user password. Use this property when creating Oracle users identified locally by the database, in other words, to indicate that the user must specify password to log on to the database


External Name – Specify External Name to create an external user. Such a user must be authenticated by an external service, such as an operating system or a third-party service. In this case, Oracle database relies on the login authentication of the operating system to ensure that a specific operating system user has access to a specific database user.

 **Important Note:** Oracle strongly recommends that you do not use external authentication with operating systems that have inherently weak login security. For more information, see Oracle Database Administrator's Guide.

Default Tablespace – Specify the default tablespace for objects that the user creates. If you omit this property, then the user's objects are stored in the database default tablespace. If no default tablespace has been specified for the database, then the user's objects are stored in the SYSTEM tablespace.

 **Restrictions:** You cannot specify a locally managed tablespace, including an undo tablespace, or a dictionary-managed temporary tablespace as a user's default tablespace.

Temporary Tablespace – Specify the tablespace for the user's temporary segments. If you omit this property, then the user's temporary segments are stored in the database default temporary tablespace or, if none has been specified, in the SYSTEM tablespace.

 **Restrictions:** The specified tablespace must be a temporary tablespace and must have a standard block size. The tablespace cannot be an undo tablespace or a tablespace with automatic segment-space management

Account Status – Specify user account status, one of the following:

- **OPEN** – user account is active and enabled
- **LOCKED** – user account is inactive and disabled.
- **EXPIRED** – user account is enabled but the password is expired; user must change his/her password on the next logon to the database
- **EXPIRED & LOCKED** – user account is inactive and disabled; in addition user password is expired. If the account status later changes to **OPEN** the password expiry status remains in affect.

Expiry Date – Date and time when the user password expired last time (expired accounts) or will expire (open accounts).

Lock Date – Date and time when the user account was locked last time. This property is automatic and cannot be changed.

Profile – Specify the default profile you want to assign to the user. The profile limits the amount of database resources the user can use. If you omit this property, then Oracle database assigns the DEFAULT profile to the user.

Consumer Group – Specify the default consumer group you want to assign to the user. If Oracle [Database Resource Manager](#) is not activated this property has no effect.

See your Oracle documentation for additional information about user account properties and how to use them.

Enabling and Disabling User Accounts

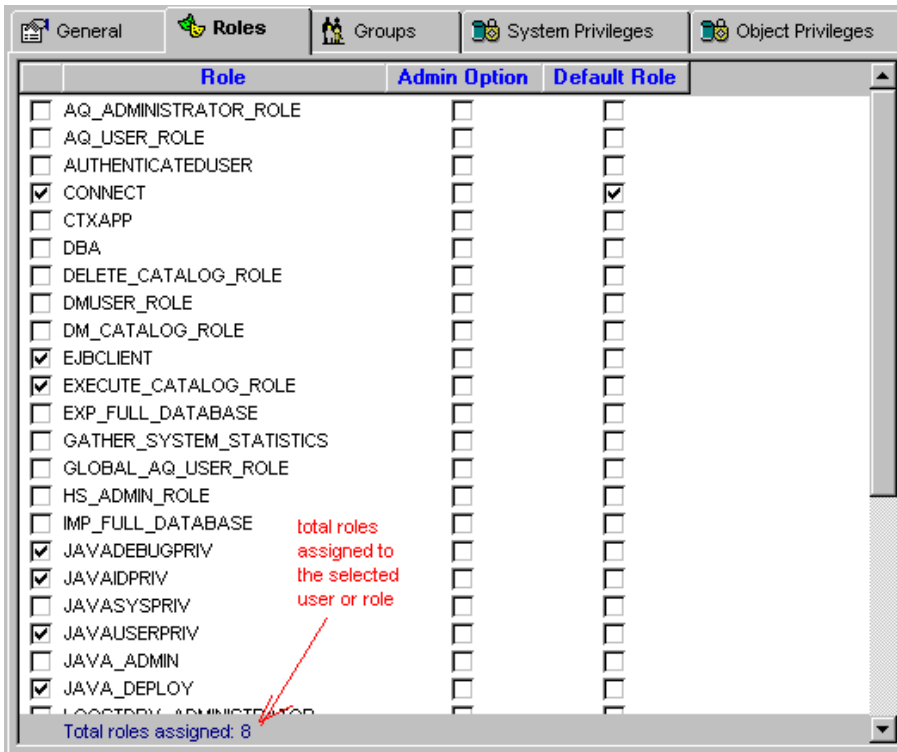
You can use the **General** page to change user account status. Select LOCKED or EXPIRED & LOCKED status from **Account Status** drop-down list to disable the user account. Select OPEN status to re-enable the user account. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

Forcing Users to Change Their Passwords

You can use the **General** page to change user account status. Select EXPIRED status from **Account Status** drop-down list to expire user password. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

Granting and Revoking Roles

The **Roles** tab page on the User Manager dialog can be used to control which roles are assigned to the selected user.



This is a multiple-choice screen. You can select as many roles as you want. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

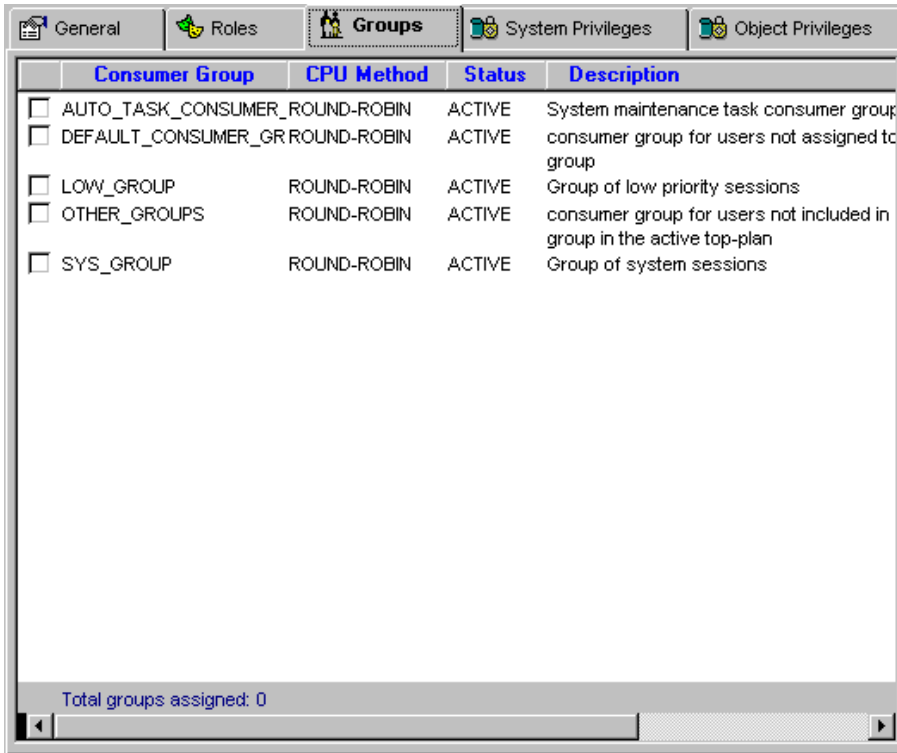
To grant a role to the selected user place a checkmark in the left-most column. To revoke a role from the selected user, clear the corresponding checkmark.

Checking the **Admin Option** box enables the user to grant this role to someone else. Checking the **Default Role** option instructs Oracle to automatically give the selected user access to the role during on connect even if the role is password-protected and normally requires the user to executing SET ROLE command in order to enable such role for the current session.

The **total** field shows how many roles are checked on the screen.


Granting and Revoking Consumer Groups

The **Groups** tab page on the User Manager dialog can be used to control which consumer groups are enabled for the selected user, in other words which groups are granted to the selected user.



This is a multiple-choice screen. You can select as many groups as you want. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

To grant a group to the selected user place a checkmark in the left-most column. To revoke a group from the selected user, clear the corresponding checkmark.

 **Tip:** Assigning a consumer group to a user is not the same as using it. It simply grants the user privileges to switch to and use the assigned group whenever that user wants. Keep in mind that every database session can be switched to only one consumer group at a time. Yet there exists a number of reasons why you may want to assign multiple groups to a single user. For example, you may want to allow the user (or database applications running under that user account) to switch to different groups and to use different resource usage plans during different times, use "LOW" usage plans for daily report processing and as opposite use "HIGH" usage plans for other mission critical business processing.

The **total** field shows how many groups are checked on the screen.

Granting and Revoking System Privileges

The **System Privileges** tab page on the User Manager dialog can be used to control which system privileges are enabled for the selected user, in other words which privileges are granted to the selected user.

Category	Privilege	Description
TABLESPACES	<input type="checkbox"/> MANAGE TABLESPACE	Take tablespaces off tablespace backups
	<input checked="" type="checkbox"/> UNLIMITED TABLESPACE	Use an unlimited amount of tablespace. This privilege overrides any specific tablespace privilege from a user, and further tablespace all specific tablespace privilege to roles.
TRIGGERS	<input type="checkbox"/> ADMINISTER DATABASE TRIGGER	Create a trigger on a database.
	<input type="checkbox"/> ALTER ANY TRIGGER	Enable, disable, or compile a trigger.
	<input type="checkbox"/> CREATE ANY TRIGGER	Create database triggers.
	<input checked="" type="checkbox"/> CREATE TRIGGER	Create a database trigger.
	<input type="checkbox"/> DROP ANY TRIGGER	Drop database triggers.
TYPES	<input type="checkbox"/> ALTER ANY TYPE	Alter object types in a schema.
	<input type="checkbox"/> CREATE ANY TYPE	Create object types in a schema.
	<input checked="" type="checkbox"/> CREATE TYPE	Create object types in a schema.
	<input type="checkbox"/> DROP ANY TYPE	Drop object types in a schema.

Total privileges granted: 5

This is a multiple-choice screen. You can select as many privileges as you want. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

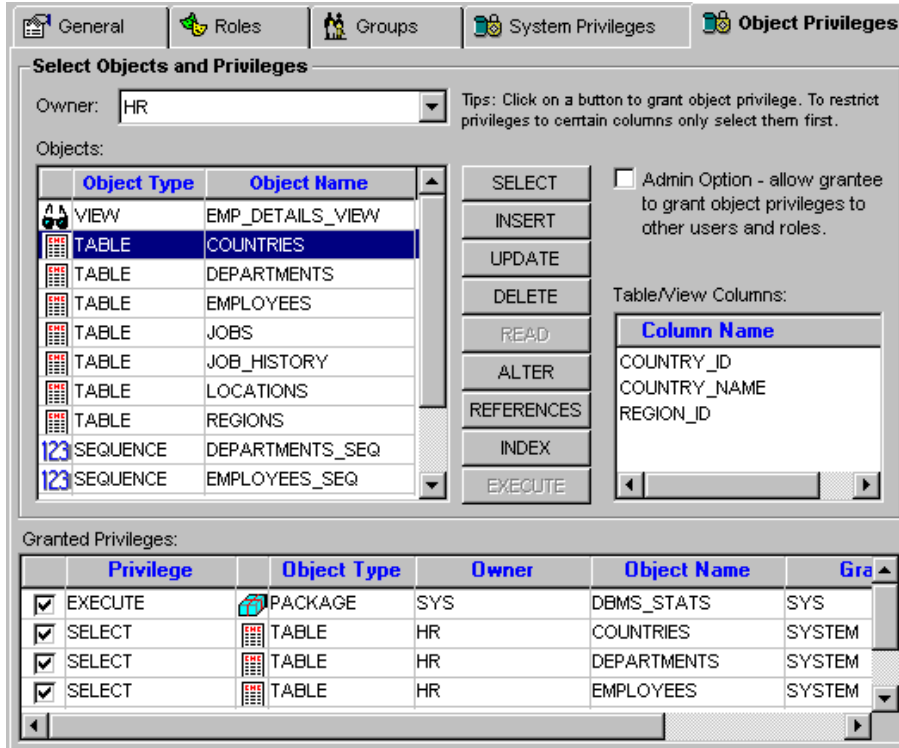
For your convenience all system privileged are grouped and sorted by category and comments explaining each privilege displayed in the right-most column.

To grant a privilege to the selected user place a checkmark in the left-most column. To revoke a privilege from the selected user, clear the corresponding checkmark.

The **total** field shows how many privileges are checked on the screen.

Granting and Revoking Object Privileges

The **Object Privileges** tab page on the User Manager dialog can be used to control which database object access privileges are enabled for the selected user, in other words which privileges are granted to the selected user.



This is a multiple-choice screen. You can select as many privileges as you want. When you click the **Apply** button all of your changes including changes on other pages are applied at once.

The **Object Privileges** tab page is split into 2 logically separate parts. The top part - **Select Objects and Privileges** is used to grant new privileges. The bottom part - **Granted Privileges** displays all object-level privileges that have been already granted to the selected user and it can be also used to revoke these privileges.

To grant a privilege to the selected user:

1. Select object's owner name (schema name) in the **Owner** drop-down list. DB Security Expert will populate the **Objects** list with all grantable objects in specified schema name.
2. Select the object you want to grant to the user in the **Objects** list. If you select a table or view DB Security Expert will populate the **Table/View Columns** list displayed on the right hand side with names of all columns available in the selected object. It will also enable/disable "grant" buttons displayed in the middle of the screen accordingly to what privileges can be granted for the selected object type.
3. if necessary check the **Admin Option** box. Checking the **Admin Option** box enables the user to grant the given privilege to someone else.
4. Click the appropriate "grant" buttons (SELECT, INSERT, and so on) to grant specific privileges to the user. These privileges will be automatically appended to the end of the **Granted Privileges** grid. In case if you made a mistake simply uncheck the left-most column in the grid for the wrong privilege.

For UPDATE, INSERT and REFERENCES operations if required you can restrict user's access to specific table/view columns only. Select all required columns in the column list then click UPDATE or INSERT or REFERENCES button once. To select multiple columns hold down the CTRL key while clicking on column names. DB Security Expert grants all-columns access to the user when no columns are selected in the column list.

To revoke a privilege from the selected user, clear the corresponding checkmark on the **Granted Privileges** grid.


Creating New Database Users

1. Open **User Manager** dialog using **Manage/Manage Users** menu.
2. Click the **New** button.
3. Fill in new user properties on the **General** tab page.
4. Enter additional options and grant permissions using other tab pages. For detailed information on how to use other tab pages read previous topics in this chapter.
5. Click the **Apply** button to save changes.


 **Tip:** After you have the new user created use [Space Usage Quotas](#) and [Resource Usage Quotas](#) tools to complete the user setup.

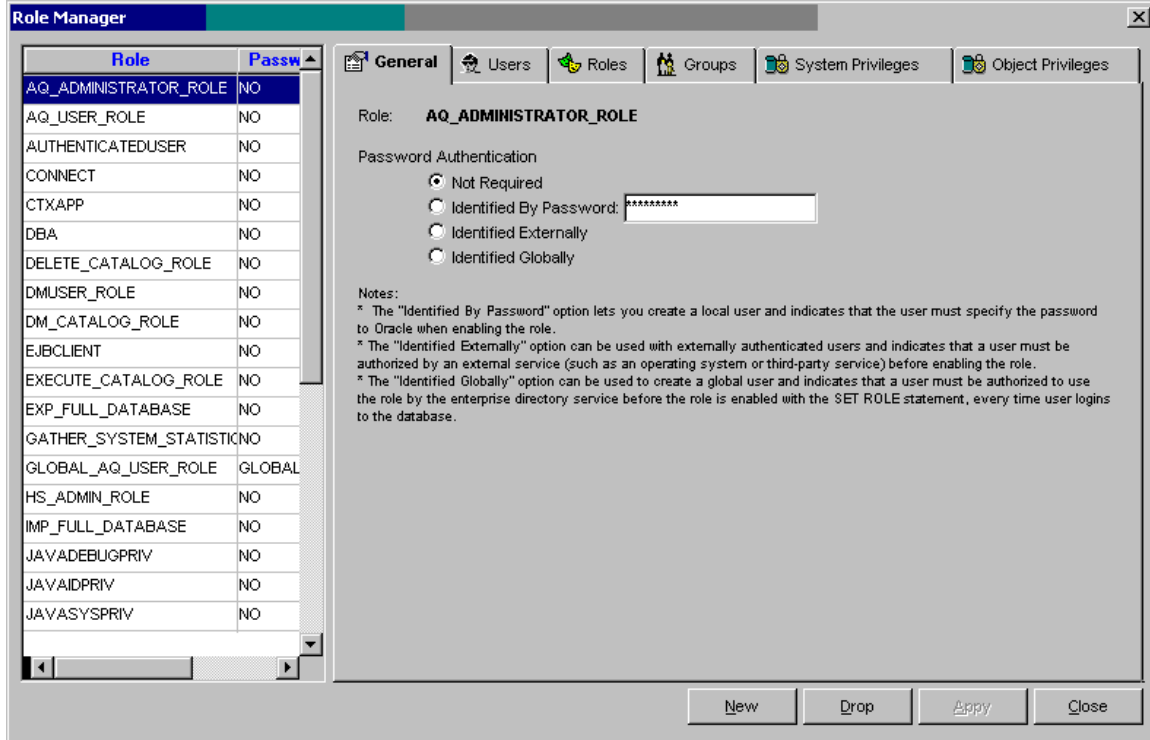
Deleting Database Users

1. Open **User Manager** dialog using **Manage/Manage Users** menu.
2. On the left hand side of the dialog select name of the user you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the YES button if you are sure that the right user is chosen for deletion.

 **Important Note:** DB Security Expert will automatically delete all user objects associated with the user schema as well as drop all referential constraints if there are any. This operation cannot be undone without restoring the user account, objects and data from a database backup.

Managing Database Roles

To start the Role Manager click **Manage/Manage Roles** menu or click the Roles  button on the application toolbar. The **Role Manager** dialog will appear.



The Role Manager dialog consists of 2 parts: The left hand side is used to list all database roles and their properties in a single table-like list. The right hand side is used to display and edit role properties assign roles to users and also assign various system-level and object-level privileges to roles. The settings and properties displayed on the right hand side are associated with the role whose name is selected in the role list on the left hand side.

You can use standard DB Tools list sorting options to rearrange how user names or properties appear in the list. For more information about supported sorting options read the [Sorting](#) topic.

Role Properties

Different Oracle database versions support different sets of role properties. On the **General** tab page DB Security Expert displays only options supported in the database you are connected to.

Role – Specify role name.

Password Authentication – Use these options to specify role password authentication.

Not Required – Choose this option to indicate that this role is authorized by the database and that no password is required to enable the role.

Identified by Password – Choose this option to create a local role and indicate that the user must specify the password to the database when enabling the role. The password can contain only single-byte characters from your database character set regardless of whether this character set also contains multi-byte characters.

Identified Externally – Choose this option to indicate that a user must be authenticated by an external service and then the user must use SET ROLE statement in order to enable this role.

Identified Globally – Choose this option to indicate that a user must be authenticated by a global enterprise directory service and then the user must use SET ROLE statement in order to enable this role.

See your Oracle documentation for more information about role properties and how to use them.

Granting and Revoking Roles to Users

The **Users** tab page on the Role Manager dialog can be used to control which users have been granted the selected role. This page is similar in functionality to the Granting and Revoking Roles page on the User Manager dialog, it is just features the opposite view of "Users granted the selected role" vs. "Roles granted to the selected user." Read [Granting and Revoking Roles](#) topic for instructions on how to use this screen.

Granting and Revoking Roles to Roles

The **Roles** tab page on the Role Manager dialog can be used to control which roles in effect have been granted the selected role. This page is similar in functionality to the Granting and Revoking Roles page on the User Manager dialog. Read [Granting and Revoking Roles](#) topic for instructions on how to use this screen.

Granting and Revoking Consumer Groups

The **Groups** tab page on the Role Manager dialog can be used to control which consumer groups are enabled for the selected role, in other words which groups are granted to the selected role.

This page is similar in functionality to the Granting and Revoking Consumer Groups page on the User Manager dialog." Read [Granting and Revoking Consume Groups](#) topic for instructions on how to use this screen.



Tip: An assigning of consumer group to a role is not the same as using it. It simply grants a privilege to switch to and use the assigned group whenever a user wants. Keep in mind that every database session can be switched to only one consumer group at a time. Yet there exists a number of reasons why may want to assign multiple groups to a single user. For example, you may want to allow the user (or database applications running under that user account) to switch to different groups and to use different resource usage plans during different times, use "LOW" usage plans for daily report processing and as opposite use "HIGH" usage plans for other mission critical business processing.

Granting and Revoking System Privileges

The **System Privileges** tab page on the Role Manager dialog can be used to control which system privileges are enabled for the selected role, in other words which privileges are granted to the selected role.

This page is similar in functionality to the Granting and Revoking System Privileges page on the User Manager dialog." Read [Granting and Revoking System Privileges](#) topic for instructions on how to use this screen.

Granting and Revoking Object Privileges

The **Object Privileges** tab page on the Role Manager dialog can be used to control which object-level privileges are enabled for the selected role, in other words which privileges are granted to the selected role.

This page is similar in functionality to the Granting and Revoking Object Privileges page on the User Manager dialog." Read [Granting and Revoking Object Privileges](#) topic for instructions on how to use this screen.

Creating New Database Roles

1. Open **Role Manager** dialog using **Manage/Manage Roles** menu.
2. Click the **New** button.
3. Fill in new role properties on the **General** tab page.
4. Enter additional options and grant permissions using other tab pages. For detailed information on how to use other tab pages read previous topics in this chapter.
5. Click the **Apply** button to save changes.


Deleting Database Roles

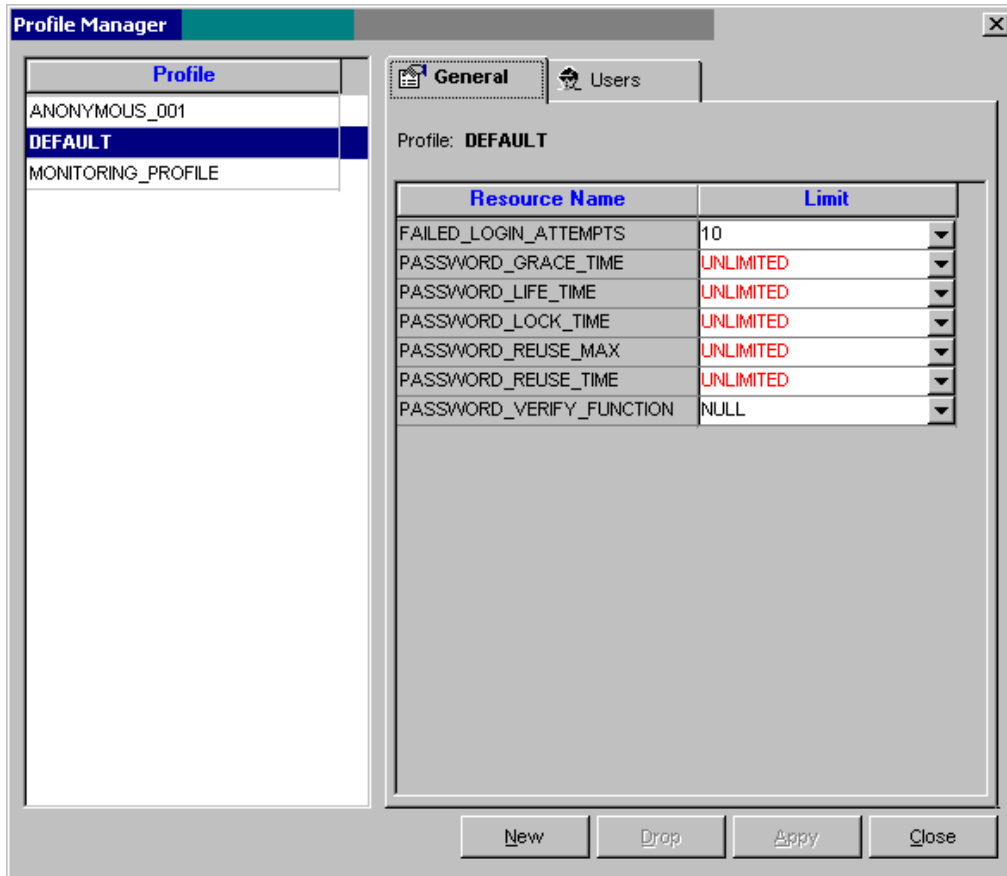
1. Open **Role Manager** dialog using **Manage/Manage Roles** menu.
2. On the left hand side of the dialog select name of the role you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the YES button if you are sure that the right role is chosen for deletion.



Important Note: All database users who have been assigned this role will automatically loose all privileges associated with this role.

Managing Database Profiles

To start the Profile Manager click **Manage/Manage Profiles** menu or click the Profiles  button on the application toolbar. The **Profile Manager** dialog will appear.



The Profile Manager dialog consists of 2 parts: The list box on the left hand side lists all database profiles. The right hand side of the dialog screen is used to display and edit profile properties and also used to assign profiles to users. The settings and properties displayed on the right hand side are associated with the profile whose name is selected in the profile list on the left hand side.

You can use standard DB Tools list sorting options to rearrange how profile names appear in the list. For more information about supported sorting options read the [Sorting](#) topic.



Tips:

- Only security related settings are displayed in profile properties. To manage resource usage-related properties use [Resource Usage Quotas](#) dialog.
- Different profile properties are supported in different Oracle versions. DB Security Expert displays only properties supported in the database you are connected to.
- Consult your Oracle documentation for detailed information about supported profile properties and their values.

Profile Properties

Database Security

FAILED_LOGIN_ATTEMPTS - Specify the number of failed attempts to log in to the user account before the account is locked.

Password Security

Parameters that set lengths of time are interpreted in number of days. For testing purposes you can specify minutes or even seconds as a decimal number. For example, to specify 5 minutes enter 0.003472 (this value was calculated $1/24/60*5$).

PASSWORD_LIFE_TIME- Specify the number of days the same password can be used for authentication. If you also set a value for **PASSWORD_GRACE_TIME**, the password expires if it is not changed within the grace period, and further connections are rejected. If you do not set a value for **PASSWORD_GRACE_TIME**, its default of **UNLIMITED** will cause the database to issue a warning but let the user continue to connect indefinitely.

PASSWORD_REUSE_TIME and **PASSWORD_REUSE_MAX** - These two parameters must be set in conjunction with each other. **PASSWORD_REUSE_TIME** specifies the number of days before which a password cannot be reused. **PASSWORD_REUSE_MAX** specifies the number of password changes required before the current password can be reused. For these parameter to have any effect, you must specify an integer for both of them.

If you specify an integer for both of these parameters, then the user cannot reuse a password until the password has been changed the password the number of times specified for **PASSWORD_REUSE_MAX** during the number of days specified for **PASSWORD_REUSE_TIME**.

For example, if you specify **PASSWORD_REUSE_TIME** to 30 and **PASSWORD_REUSE_MAX** to 10, then the user can reuse the password after 30 days if the password has already been changed 10 times.

If you specify an integer for either of these parameters and specify **UNLIMITED** for the other, then the user can never reuse a password.

If you specify **DEFAULT** for either parameter, then Oracle Database uses the value defined in the **DEFAULT** profile. By default, all parameters are set to **UNLIMITED** in the **DEFAULT** profile. If you have not changed the default setting of **UNLIMITED** in the **DEFAULT** profile, then the database treats the value for that parameter as **UNLIMITED**.

If you set both of these parameters to **UNLIMITED**, then the database ignores both of them.

PASSWORD_LOCK_TIME - Specify the number of days an account will be locked after the specified number of consecutive failed login attempts.

PASSWORD_GRACE_TIME - Specify the number of days after the grace period begins during which a warning is issued and login is allowed. If the password is not changed during the grace period, the password expires.

PASSWORD_VERIFY_FUNCTION - Specify the name of the password complexity verification routine. Oracle database provides a default script, but you can create your own routine instead. Specify **NULL** to indicate that no password verification is performed.

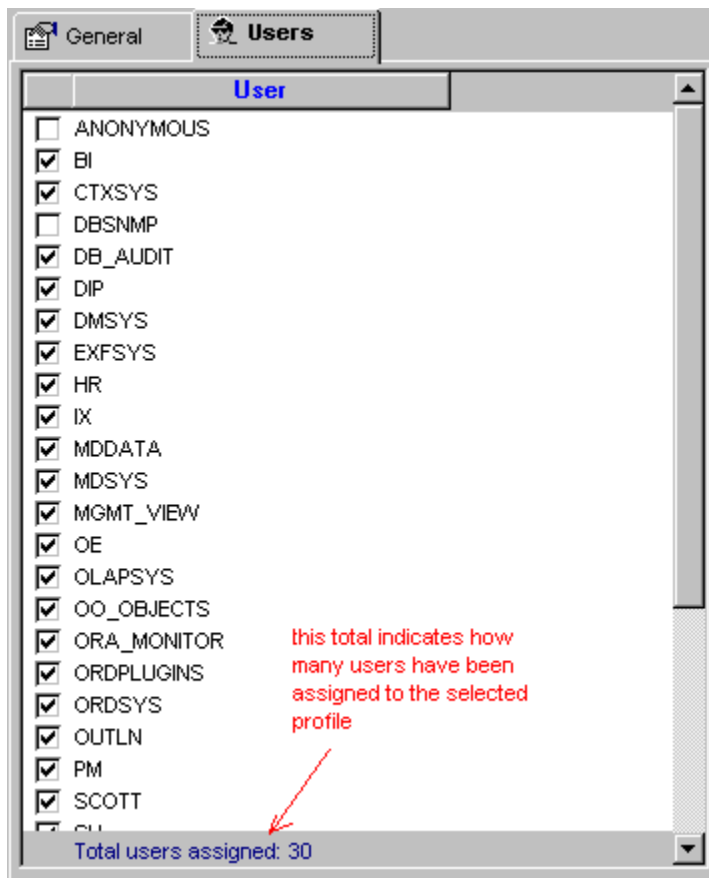


Tip: **PASSWORD_VERIFY_FUNCTION** is only **used to verify password complexity** at the time of the user account creation or password change. It is **NOT USED for the user authentication** at the time of user logon to the database.

Working with Profiles

To modify an existing profile:

1. Start the **Profile Manager**.
2. Select the required profile in the profile list.
3. On the **General** tab page modify profile properties as needed
4. On the **Users** tab page modify user assignments as needed.



This is a multiple-choice screen. You can select as many users as you want. When you click the **Apply** button all of your changes including all changes on **General** page are applied at once.

5. To assign a user to the selected profile place a checkmark in the left-most column. To undo the assignment clear the corresponding checkmark.

Note: User is automatically reverted to the DEFAULT profile if the previous profile assignment is cleared.

6. Click the **Apply** button to save changes.

To delete an existing profile:

1. Start the **Profile Manager**.
2. In the profile list select name of the profile you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the YES button if you are sure that the right profile is chosen for deletion.

Important Note: **The DEFAULT profile may not be deleted!**

To create a new profile:

1. Start the **Profile Manager**.
2. Click the **Add** button
3. On the **General** tab page modify profile properties as needed.
4. On the **Users** tab page assign users to the new profile as needed.
5. Click the **Apply** button to save changes.

DB Log Expert

Overview

How DB Log Analyzer Can Help

DB Log Expert can help DBAs and developers in monitoring existing databases and balancing database loads. In addition, it provides graphical views of the current redo-log configuration, including statuses and usage information for all redo-log members and redo-log files. By presenting information graphically, DB Log Expert allows DBA to quickly identify overall database problems and recognize patterns of the database workload.

Before You Begin

You must have SELECT privileges on the Oracle dynamic performance tables. If you do not have sufficient authority to use DB Log Expert, contact your database administrator. Oracle8 users also need permissions to use DBMS_SQL package.

Redo Log Configuration and Status

This report shows current database configuration of Redo Log Groups and its Members. Note that all members of the same redo log group are displayed on the same line. You should see as many lines as many redo log groups are configured in the database.

What you can do:

- Click on the Refresh button to refresh report data.
- Click on the Print button to print the graph.
- Click on the Archive Now button to force immediate archiving of all redo logs. You must have sufficient authority to perform archival operation.
- Click on the Switch Log button to switch current redo log. You must have sufficient authority to perform this operation.

Redo Log Latches

This report analyzes and displays redo log performance statistics. If redo latches are detected, the report provides expert advise on how to avoid them

What you can do:

- Click on the Refresh button to refresh report data.
- Click on the Print button to print the report.

Database Load History Report

This report shows yearly and monthly database load statistics based on the number of database changes recorded in the Redo Log Files

What you can do:

- Use Year drop-down list box to change the report year.
- Use Month drop-down list box to change report month of the year chosen in the Year drop-down list box. Alternatively, to change the selected month, you can double-click on the required month bar (column) in the Year statistics graph.
- Click on the Print button to print the Monthly graph.

Database Load Weekly Analysis – Day by Day Comparison

This report shows your database load by day of week and time. For your convenience statistics are grouped by hour and date. Daily tables are displayed side by side so you can easily see load patterns and trends.

What you can do:

- Click on the Refresh button to refresh report data.
- Click on the Print button to print the graph.

Checkpoints Monitor

This is the real-time monitor for tracking database checkpoints as they occur.

What you can do:

- Use the Interval field to set required timer interval (refresh rate). To turn off automatic refresh enter zero.
- Click on the Refresh button to manually trigger new refresh.
- Click on the Print button to print the graph.
- Click on the Checkpoint button to force new database checkpoint. You must have sufficient authority to perform this operation.

Database Load Trends Report

This report allows you to explore historical patterns of the changes made in your database. DB Log Expert currently provides two report views:

1. Activity Level by Day of Week
2. Activity Level by Time (Hourly)

What you can do:

- Use Scale drop-down list box to change Trends graph scale.
- Select from the Start Year drop-down list box start year for the analysis.
- Select from the Start Month drop-down list box start month for the analysis.
- Click on the Print button to print the graph.

DB Alert Expert

Overview

DB Alert Expert provides a way to send various information messages from your Oracle databases. DB Alert Expert is targeted primarily to Oracle DBAs and developers.

Oracle DBAs can use DB Alert to implement advanced database monitoring solutions. Whenever database requires some attention it can use DB Alert functions to alert DBAs, operators, and other people using email, network messages, or pager. For example, a database deadlock monitoring job can automatically send an email alert to DBA notifying about the problem. Another good example is a database free space monitoring job, which can notify DBAs when a tablespace is low of space and a new file is needed for that tablespace. This can save a lot of DBA time as he/she/ won't be required anymore to manually run various scripts to check critical database parameters on a regular bases.

With the help of DB Alert Oracle developers can easily add messaging functions to their homegrown applications. For example, an application can automatically alert sales people when the inventory drops below the critical level, or an office management application can send an alert message to all users reminding them to submit their attendance timesheets.

DB Alert automatically logs all messages passed from your databases. Therefore, it can also serve as a valuable troubleshooting tool for developing, setting up and testing new applications. This option is especially helpful when developing and testing back-end applications that have no user interface.

DB Alert comes with two ready to use monitoring packages that you can install using DB Tools [Database Setup utility](#):

- **Log Monitor** – The primary purpose of this package is to enable automated monitoring of Oracle Alert Log files. It can be also used to monitor other files such as trace files and log files produced by various batch jobs running on the same server or shared drive/device. For more information about the Log Monitor see [Monitoring Alert Log Files and Trace Files](#) topic.
- **Space Monitor** - The purpose of this package is to automate free-space monitoring in tablespaces and alert DBA if the amount of free space falls below the specified threshold. Each tablespace may have its own threshold set to an appropriate level. For more information about the Space Monitor see [Monitoring Tablespaces](#) topic.



Tip: You can easily email-enable your own back-end monitoring functions by simply calling DB_ALERT function with the required message parameters.

DB Alert Expert supports the following communication methods:

- **Email using Windows Messaging Application Interface (MAPI)**
- **Email using Simple Mail Transfer Protocol (SMTP)**
- **Email using Lotus Notes**
- **Sending network messages using NET command**
- **Paging to alphanumeric pager using email-to-pager service (not provided with DB Alert)**

DB Alert is a powerful multi-threaded application that can be configured to simultaneously serve multiple Oracle databases residing on the same or different computers that are connected to the same network. Because DB Alert uses SQL*Net for all database connections, it can connect to any Oracle database system running on any platform.

DB Alert runs a separate daemon process for every configured database system. In the event if one database goes down, it does not stop/prevent DB Alert from serving other databases. That daemon process instantly checks for 'DB_ALERT' messages sent using DBMS_PIPE procedure. When a message is found in

the pipe, the daemon removes that message from the pipe, writes it to the log and then optionally emails or pages it to the recipients specified in DB Alert options.

DB Alert Installation and Usage

You can use default DB_ALERT function provided with DB Tools (use the graphical [Database Setup utility](#)) or code your own function or method to write 'DB_ALERT' messages to the database pipe. The graphical [Database Setup utility](#) will create ORA_MONITOR schema and DB_ALERT function that can be used to send messages to DB Alert. Note that both DB_ALERT function and DB Alert daemon processes use DBMS_PIPE package to communicate. You can also create and use DB_ALERT or similar function in your own schema. Here is the code for such function.

```
CREATE OR REPLACE FUNCTION db_alert( severity VARCHAR2, message VARCHAR2 )
return NUMBER
IS
    -- severity must be one of the following: 'INFO', 'WARNING', 'ERROR'
    -- message can be any string up to 2000 characters long

    -- return code is the value returned by dbms_pipe.send_message() function
    -- for supported return codes please see Oracle PL/SQL Guide for
    -- dbms_pipe.send_message function
BEGIN
    dbms_pipe.pack_message(severity);
    dbms_pipe.pack_message(message);

    RETURN dbms_pipe.send_message('DB_ALERT');
END;
```

The following example demonstrates how to send a message via DB Alert:
This can be done in both SQL and PL/SQL code

```
SELECT db_alert( 'WARNIING', 'Low of space in tablespace TOOLS' ) FROM dual;
```

Example fragment of PL/SQL code

```
DECLARE
    number rc;
BEGIN
    -- . . . some PL/SQL code here . . .
    rc := db_alert( 'INFO, 'Please do not forget to submit your ' ||
        'weekly timesheet before 5:00 PM today' );
    -- . . . some PL/SQL code here . . .
END;
```

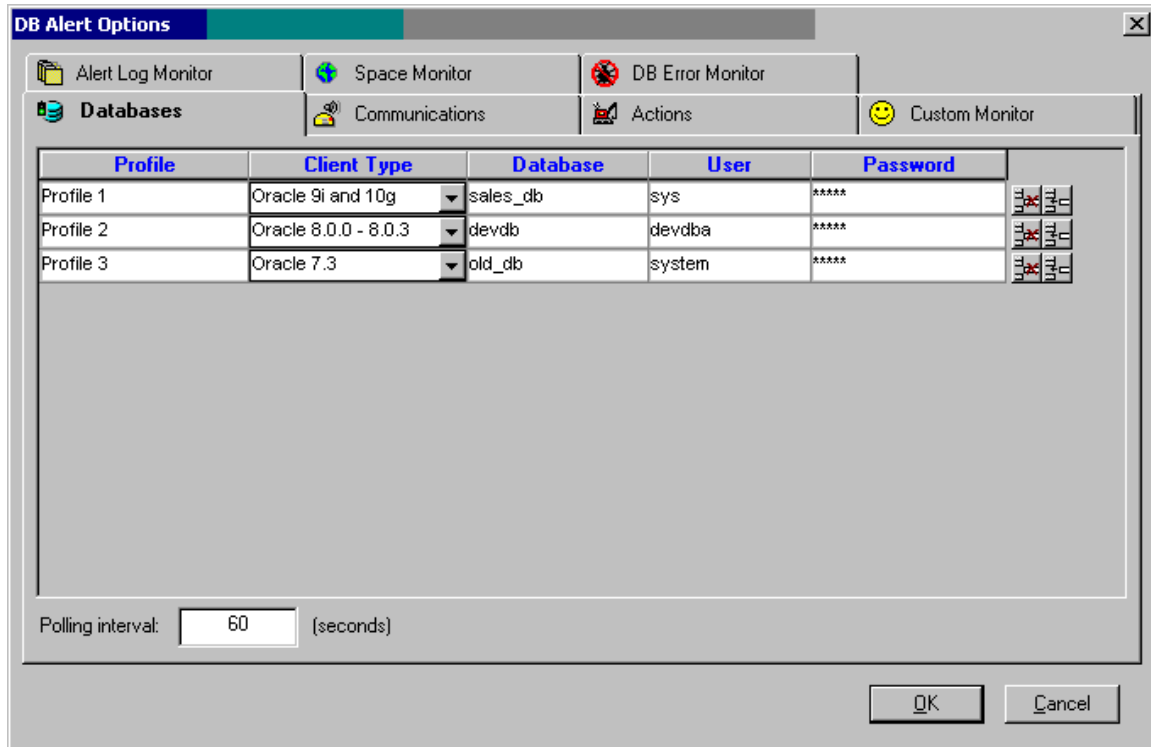
See also:

[Configuring DB Alert Expert](#) Starting DB Alert Expert on Computer Startup

Configuring DB Alert

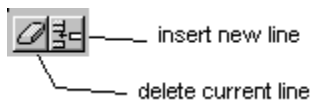
Before you can use DB Alert you need to configure at least three DB Alert areas: database connections, messaging interface and properties, and alert processing methods. If you have installed Alert Log Monitor and Free Space Monitor packages you can also use DB Alert Options screen to configure settings for these packages. To open DB Alert Options dialog select **File/Options** in the DB Alert menu. A tab page on the DB Alert Options dialog presents each group of settings.

Databases



Here you specify database connections for DB Alert. Before you add new database connection, make sure you have already created DB_ALERT or similar function in that database. To create this function you can use the graphical [Database Setup utility](#) provided with DB Tools or code your own method. See the [DB Alert Overview](#) topic for details and examples for coding and using such functions.

Make sure that user whose ID you use to connect DB Alert to the database, has EXECUTE permissions on the standard DBMS_PIPE package.



Use Delete and Insert buttons to add and delete database profiles

For details on setting up database connection parameters see [How to Connect to Your Database](#) topic.

Communication Methods

The screenshot shows the 'DB Alert Options' dialog box with the 'Communications' tab active. The 'Send Method' section has radio buttons for 'MAPI email interface' (selected), 'SMTP email interface', 'Lotus Notes email interface', and 'Network messages'. The 'MAPI Properties' section includes a 'MAPI Profile' dropdown set to 'Exchange Settings', a 'Password' field with 'xxxxxxx', and a 'Send' dropdown set to 'By Name'. The 'SMTP Properties' section includes 'SMTP Server' (mycompany.com), 'Port' (25), 'Encoding' (mime), and 'Sender's email address' (dba@mycompany.com). The 'Lotus Notes Properties' section has empty 'Logon User' and 'Password' fields. 'OK' and 'Cancel' buttons are at the bottom right.

DB Alert supports three different email interfaces: standard Windows MAPI (Messaging Application Programming Interface) interface, SMTP (Simple Mail Transfer Protocol), and Lotus Notes interface utilizing Lotus Notes API. DB Alert uses the selected interface when sending various alert messages.

DB Alert also supports network messages. Network messages like email messages can be sent to individual people or to a group of people specified by the group name. This communication method uses the NET command to send messages. Run "NET HELP SEND" from the command prompt for help with the NET command.


If your company supports an email-to-pager service you can also configure DB Alert to page messages to alphanumeric pagers and cell phones by sending such messages to certain email addresses. In many instances cell phone or pager's email addresses have simple < phone-number>@<phone-company.com> format.



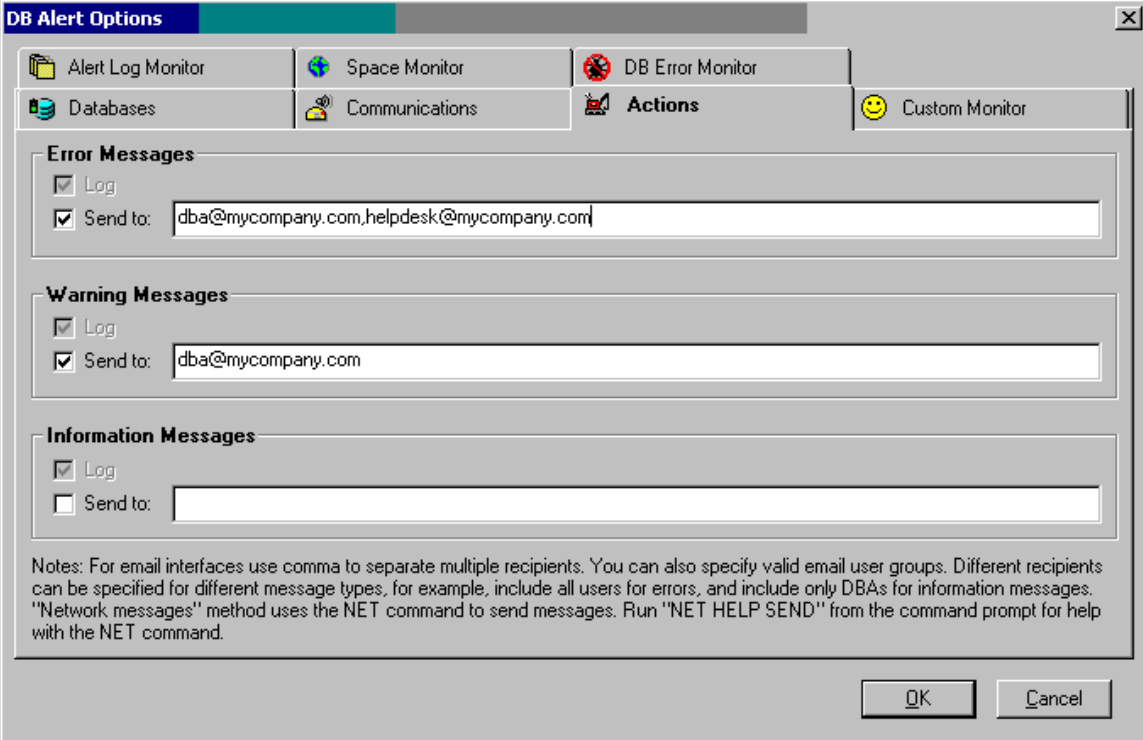
Important notes:

- MAPI and SMTP interfaces enable DB Alert to interact with multiple messaging systems across a variety of software and hardware platforms whereas Lotus Notes interface works with the following configurations only:
 - 1) Lotus Notes workstation v4.5 and above running on Windows NT workstation or server (Intel platforms only). There is no limitation on the Lotus Notes server version and platform.
 - 2) Lotus Notes server v4.5 and above running on Windows NT server (Intel platform only).
- If you have installed Lotus Notes MAPI extensions, you can still use MAPI interface to send and receive email via Lotus Notes. We highly recommend using MAPI interface whenever possible, because (due to nature of Lotus Notes API) we cannot guarantee that Lotus Notes interface will be compatible with the future versions of Lotus Notes. However, we will try to release updated versions of Lotus Notes interface after new versions of Lotus Notes become available.
- DB Alert Lotus Notes interface consists of two parts:
 - Notes email interface library
 - Notes extension manager

DB Alert email interface library for Lotus Notes always uses default Notes mail database and mail server. Default settings are taken from Notes environment variables.

 **Warning:** DB Alert installs Notes extension manager program that intercepts the Notes password prompt, and supplies the password that you specified for an email in the DB Alert. The extension manager allows sending email messages via Notes without user intervention when Notes normally requires a password. The extension manager program is built as a set of dynamic link libraries (DLLs). These DLLs are loaded by Lotus Notes on startup, and they behave as if they are part of the Lotus Notes software. While DB Alert email operation is in progress, your Lotus Notes is exposed to other users and programs because no password is required at that moment to interact with the Lotus Notes software. Before selecting Lotus Notes interface, make sure you don't have another Notes extension manager already installed on your system. To verify this, make sure you don't have EXTMGR_ADDINS key in the NOTES.INI file or that key is not initialized.

Alert Actions



DB Alert Options

Alert Log Monitor | Space Monitor | DB Error Monitor

Databases | Communications | **Actions** | Custom Monitor

Error Messages

Log

Send to: dba@mycompany.com,helpdesk@mycompany.com

Warning Messages

Log

Send to: dba@mycompany.com

Information Messages

Log

Send to:

Notes: For email interfaces use comma to separate multiple recipients. You can also specify valid email user groups. Different recipients can be specified for different message types, for example, include all users for errors, and include only DBAs for information messages. "Network messages" method uses the NET command to send messages. Run "NET HELP SEND" from the command prompt for help with the NET command.

OK Cancel

DB Alert supports three different action types:

- ERROR
- WARNING
- INFO

You specify action type as a parameter to DB_ALERT or similar function when send alerts from within your database. To create this function you can use the graphical [Database Setup utility](#) provided with DB Tools or code your own method. See the [DB Alert Overview](#) topic for details and examples for coding and using such functions.

By default all alerts for all action types are logged to the screen and can be optionally printed. For every message type you can optionally specify whether you want messages of that type to be send to one or more recipients using the selected communication method.

For email interfaces use comma to separate multiple recipients. You can also specify valid email user groups. Different recipients can be specified for different message types, for example, you can include all users for errors, and include only DBAs for information messages.

Like email messages described above, network messages can be sent to individual people or to a group of people specified by the group name. Such messages are sent using the NET command. Run "NET HELP SEND" from the command prompt for help with the NET command.

Log Monitor Options

This area is used to configure Log Monitor options.

DB Alert Options

Databases | Communications | Actions | Custom Monitor

Alert Log Monitor | Space Monitor | DB Error Monitor

1. Select database server
 sales_db (Profile 1)

2. Add/delete name of alert log files. Specify just file names without path. Files will be searched in the directory specified in BACKGROUND_DUMP_DEST parameter. Make sure your <INIT.ORA> contains UTL_FILE_DIR parameter with the same value as in BACKGROUND_DUMP_DEST. Without this parameter Alert Log Monitor will be unable to access specified files.

File Name
ora81alert.log

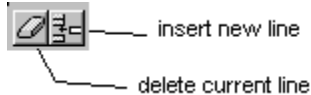
3. Specify messages to be monitored in the specified log files. Search text can include % and _ symbols for LIKE searches.

Severity	Text to search in log files:
WARNING	%Error%
ERROR	%ORA-01013%


OK Cancel

To configure the required database:

1. Make sure the database profile is complete on the **Databases** tab page
2. Select the profile name from the **Database** drop-down box
3. Enter names of files that you want to monitor. Use Delete and Insert buttons to add and delete lines in the **Files** box.



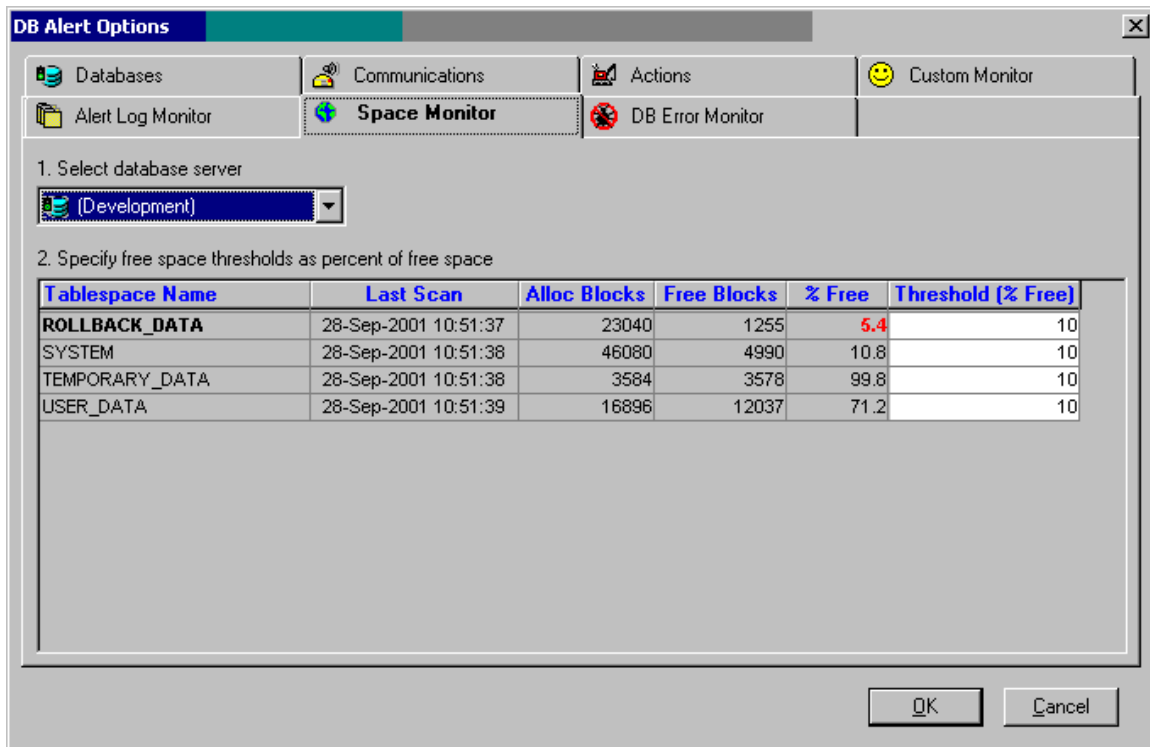
4. Enter complete strings or search masks to be searched in the specified files. To specify a search mask you can use standard Oracle % methods for character substitution. Use Delete and Insert buttons to add and delete search lines to the **Search Strings** box.

 **Tip:** Messages may have different severity. Please note that the **Severity** option is used not to indicate how important the message is to you, but to indicate **which action to perform** when such message is found. Actions to be performed you actions on the **Actions** tab.

5. If necessary, select another database profile and repeat steps 2 to 4.

Space Monitor Options

This area is used to configure Space Monitor options.

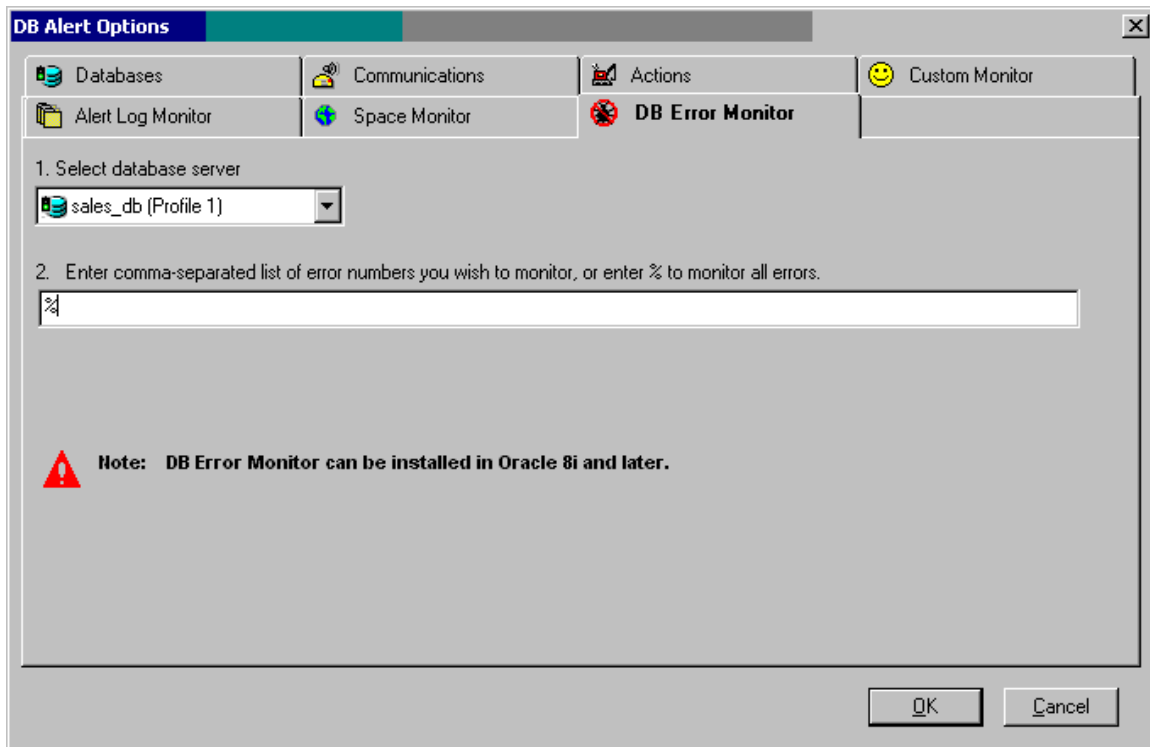


To configure the required database:

1. Make sure the database profile is complete on the **Databases tab** page
2. Select the profile name from the **Database** drop-down box
3. Specify required free space threshold as a percent of free space.
4. If necessary, select another database profile and repeat step 3.

Database Error Monitor Options

This area is used to configure Log Monitor options.



To configure the required database:

1. Make sure the database profile is complete on the **Databases tab** page
2. Select the profile name from the **Database** drop-down box
3. Enter comma-separated list of database errors for which you want to be notified by DB Alert. To monitor all database errors enter percent sign into the **Errors** field.
4. If necessary, select another database profile and repeat step 3.


Starting DB Alert on Computer Startup

To start DB Alert each time Windows starts:

1. Click the Windows **Start** button, and then click the **Settings** menu.
2. Click **Taskbar**, and then click the Start Menu **Programs** tab.
3. Click **Add**, and then click **Browse**.
4. Locate DB_TOOLS.EXE file in the DB Tools installation directory and double-click it.
5. Click the **Next button** and then double-click the **StartUp** folder.
6. Type the name **DB Alert**, which you will later see in the **StartUp** menu, and then click **Finish**. Windows will create a shortcut that will be placed to the **SratUp** folder.
7. Repeat steps 1 and 2. Click **Advanced**, and then locate the newly created shortcut.
8. Right-click the shortcut, and then click **Properties**
9. Add "/ALERT" command line parameter for the DB_TOOLS.EXE. The complete command line should look like the following:

"C:\Program Files\DB Tools 5\db_tools.exe" /ALERT

10. For the startup window property select **Run Minimized**, and then click **OK**

When DB Alert will start on next Windows start, it will appear as an icon  in the Windows System Tray. To show DB Alert console, simply double-click on the icon.

Monitoring Alert Log Files and Trace Files

The **Log Monitor** package can be used to monitor the Oracle alert log and other text files for error messages and raise the alert whenever a new message is found.

The importance of automated alert log monitoring is very high. The alert log file lists significant things that happen to your database. It alert log also records errors that you might not notice for a while, like *ORA-01631: max # extents reached in table/cluster XXX* or *ORA-00600, block corruption error*.

Many DBAs examine the log every day and read logs on the screen or use some scripts to scan it for error messages. If you are responsible for a number of nodes and a number of instances, this can be a tedious exercise. Another issue with this approach is that if you run a script, it is not enough for the script to simply check the alert file for errors. The script needs to know if errors were written to the alert file between the times that it last checked and now. To accomplish this, the script must somehow remember what the previous version of the alert file looked like. All this is done in the Log Monitor package provided with DB Tools. The Log Monitor package uses DB Alert **send_message** function to notify personal about found problems.

To install and configure Log Monitor:

1. Use DB Tools [Database Setup utility](#) to install the package and schedule log monitoring job.
2. Use DB Alert Expert console options (File/Options menu) to specify files and message you want to monitor. Please see [DB Alert Configuration](#) topic for details.



Tip: The installation routine by default creates new job that executes Log Monitor analysis function every 15 minutes. You can adjust job settings or completely disable this job using [DB Job Expert](#).

Monitoring Free Space in Tablespaces

Amount of free space in tablespaces is one of the Oracle database essential measurements that every DBA should monitor. Oracle returns an error when a tablespace does not contain enough free space for Oracle to dynamically allocate an extent to a requesting table or index (*ORA-1547 Cannot allocate new extent in XXX* and *ORA-1631: max # extents reached in table/cluster*). For example, if a tablespace contains 10M of free space but a table or index in that tablespace requests an extent with the size greater than 10M, Oracle will return an error to the session attempting to insert a row into that table. The DBA may easily avoid such errors if the DBA monitors tablespace free space on a regular basis. The **Space Monitor** package is designed specifically for this purpose. It can be used to monitor free space and raise an alert whenever the amount of free space falls below the predefined threshold.

The Space Monitor package uses DB Alert **send_message** function to notify personal about found problems.

To install and configure Space Monitor:

1. Use DB Tools [Database Setup utility](#) to install the package and schedule space-monitoring job.

2. Use DB Alert Expert console options (File/Options menu) to specify files and message you want to monitor. Please see [DB Alert Configuration](#) topic for details.



Tip: The installation routine by default creates new job that executes Space Monitor analysis function every 30 minutes. You can adjust job settings or completely disable this job using [DB Job Expert](#).

Monitoring Database Errors

The **Monitor Database Errors** option can be used to monitor all kinds of database errors and raise automatic whenever new errors occur.

This option is only available in Oracle 8i and later. In Oracle versions 8i and later DB Alert can take advantage of Oracle system-level triggers to capture Oracle server errors. DB Alert can audit all or specific Oracle server errors. Detected errors can be then automatically emailed to you.

To install and configure Error Monitoring use either of the following:

1. Use DB Alert Expert console options (File/Options menu) to specify errors you want to monitor. Click the OK button. DB Alert will prompt for your permissions to you to install the required objects. Click YES button to confirm.
2. Use DB Tools [Database Setup utility](#) to install the necessary DB alert procedures and system triggers.



Tip: The installation routine creates ORA_MONITOR.ALERT_TR_ERRORS trigger ON DATABASE ERROR event. You must have CREATE ANY TRIGGER privileges in order for this operation to complete successfully.

Please see [DB Alert Configuration](#) topic for details on how to configure Database Error Monitoring options.

DB Audit Expert

Overview

DB Audit Expert combines two sets of auditing tools: tools for the system usage auditing and tools for the data change auditing. First set of tools provides you with easily navigated graphical interface to the powerful Oracle system auditing features where second set of tools allows you to setup data auditing which is not supported by Oracle database software directly. DB Audit Expert also provides you with several pre-built auditing reports.

See also:

- [System Auditing](#)
- [Data Change Auditing](#)

System Auditing

DB Audit Expert provides a rich graphical interface for Oracle system audit management and audit data analysis. All Oracle database systems support three standard types of auditing: namely SQL statement-level, privilege-level, and object-level auditing. Audit records can be written to the standard Oracle audit table SYS.AUD\$ or to an external file. The three basic types of auditing can be performed by user, successful or unsuccessful attempts, and by session or access time intervals. The standard system auditing does not include data-change auditing at the record or column level. DB Audit provides this missing functionality using advanced data-change audit methods and tools that work at the record and column levels. To learn more about data-change auditing in Oracle see [Data Change Auditing](#) topic.

Through the use of the standard auditing capabilities it is possible to audit virtually any system-event such as:

- User logon/logoff times and activity
- Execution of a specific SQL statement, such as auditing table manipulation operations
- Use of system privilege(s), there are over 100 privileges such as CREATE VIEW, SELECT ANY TABLE, etc.
- Audit by object, to see what was done to an object (tables, views, indexes, etc.).
- Audit specific user(s).
- Audit middle tier applications, such as access by another database
- Audit successful and/or unsuccessful events, including unsuccessful logons and object access.



Notes:

- The system-level database auditing is performed entirely on the server. DB Audit Expert uses Oracle DBMS SQL commands to access the system audit trail information and manage system audit configuration and settings. To enable auditing at any level, you will need to set the AUDIT_TRAIL=DB parameter in the instance parameters INIT.ORA file and then restart the instance: this is not settable dynamically. On UNIX based systems this file is normally located in the 'dbs' subdirectory under the ORACLE_HOME directory. On Windows based systems this file is named INITORCL.ORA and it is normally located in the 'Database' subdirectory under the ORACLE_HOME directory.
- You must have AUDIT ANY and CREATE TABLE privileges.

System Auditing Considerations

Although system auditing is relatively inexpensive, limit the number of audited events as much as possible. This will minimize the performance impact on the execution of statements that are audited, and minimize the size of the audit trail. Consider the following two general guidelines when devising an auditing strategy:

1. Evaluate your purpose for auditing. Once you have a clear understanding of the reasons for auditing, you can devise an appropriate auditing strategy and avoid unnecessary auditing. For example, suppose you are auditing to investigate suspicious database activity. This information by itself is not specific

enough. What types of suspicious database activity do you suspect or have you noticed? A more focused auditing purpose might be to audit unauthorized deletions from arbitrary tables in the database. This purpose narrows the type of action being audited and the type of object being affected by the suspicious activity.

2. Audit knowledgeable. Audit the minimum number of statements, users, or objects required to get the targeted information. This prevents unnecessary audit information from cluttering the meaningful information and consuming valuable space in the SYSTEM tablespace. Balance your need to gather sufficient security information with your ability to store and process it. For example, if you are auditing to gather information about database activity, determine exactly what types of activities you are tracking, audit only the activities of interest, and audit only for the amount of time necessary to gather the information you desire. Do not audit objects if you are only interested in each session's logical I/O information.

Because volume of audit trail information usually grows very fast you should regularly archive audit records and purge the audit trail.

Once you have collected the required information, archive the audit records of interest and purge the audit trail of this information.

Before You Begin System Auditing

You must have AUDIT ANY, CREATE TABLE privileges.

Archiving system audit trail to a table

1. Select **Audit/System Audit Trail/Archive to Table** command from the DB Audit Expert menu.
2. The select destination table dialog appears. Specify either name of an existing table to which you archived before using DB Audit Expert or enter the name of a new table. DB Audit Expert automatically creates the destination table if it does not exist. If you select an existing table, then DB Audit Expert appends data from the system audit trail to that table archived. Note that if have not purged the system audit trail since last archiving operation, you will get the same audit trail records archived twice

Archiving system audit trail to a file

1. Select **Audit/System Audit Trail/Export to File** command from the DB Audit Expert menu.
2. The select export file dialog appears. Specify name of the file to which you want to archive the system audit trail. If you select an existing file, DB Audit Expert will overwrite this file.

Purging system audit trail

1. Select **Audit/System Audit Trail/Purge** command from the DB Audit Expert menu.

System Audit Options

- Depending on the auditing options set, audit records can contain different types of information. However, all auditing options generate the following information:
- The user that executed the audited statement
- The action code (a number) that indicates the audited statement executed by the user

- The object or objects referenced in the audited statement
- The date and time that the audited statement was executed

To select statements, objects, and users for auditing, use **Audit/System Audit Trail/Set Audit Options** command from the DB Audit Expert menu. Follow instructions provided by the System Audit Options Wizard.

For more information on the system audit trail and system audit options see your Oracle documentation.

Advanced System Auditing Options

Automatic Audit Trail Space Management

Moving Oracle native system audit trail table out of the SYSTEM tablespace.

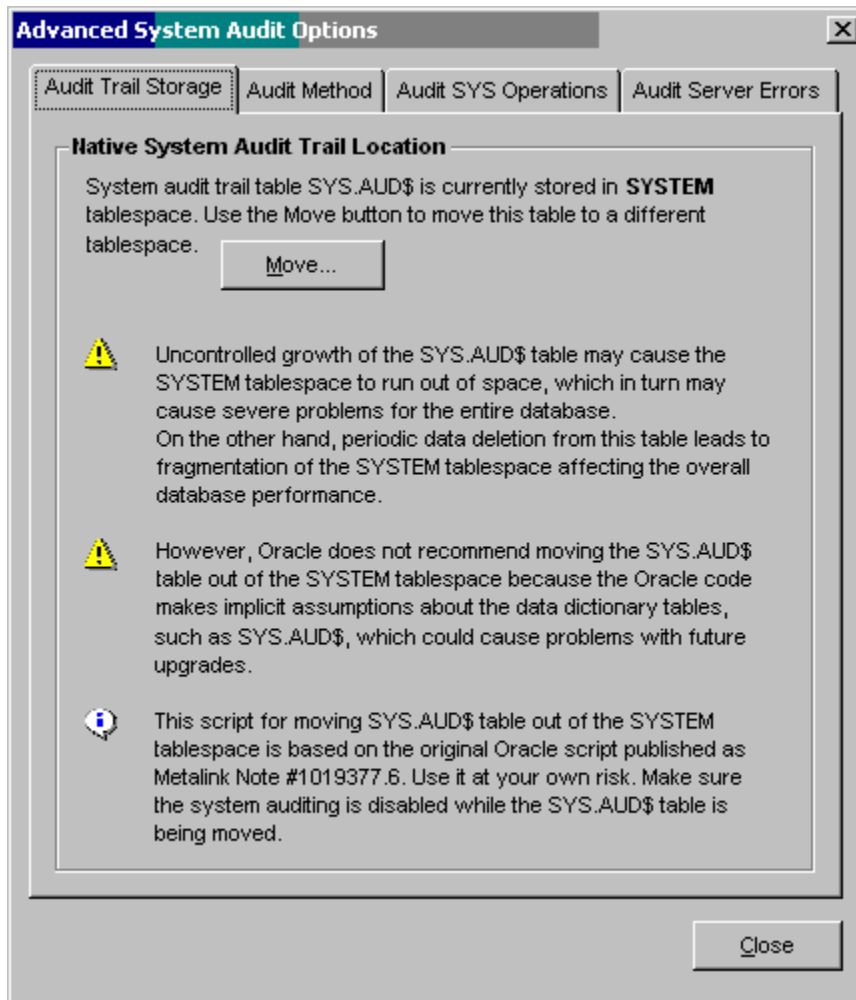
The amount of output generated by Oracle system audit can be quite large. All system audit data is stored in a single table SYS.AUD\$ stored as all Oracle SYS tables in the database SYSTEM tablespace.

Uncontrolled growth of SYS.AUD\$ table may cause the SYSTEM tablespace to run out of space, which in turn may cause severe problems for the entire database - the database would hang if all the free space in the SYSTEM tablespace were to be used up by audit records, that's why it is important to move the audit table SYS.AUD\$ from the SYSTEM tablespace.

Periodically deleting old data from the SYS.AUD\$ table is not a very good solution to the space problem because it leads to fragmentation of the SYSTEM tablespace affecting the overall database performance.

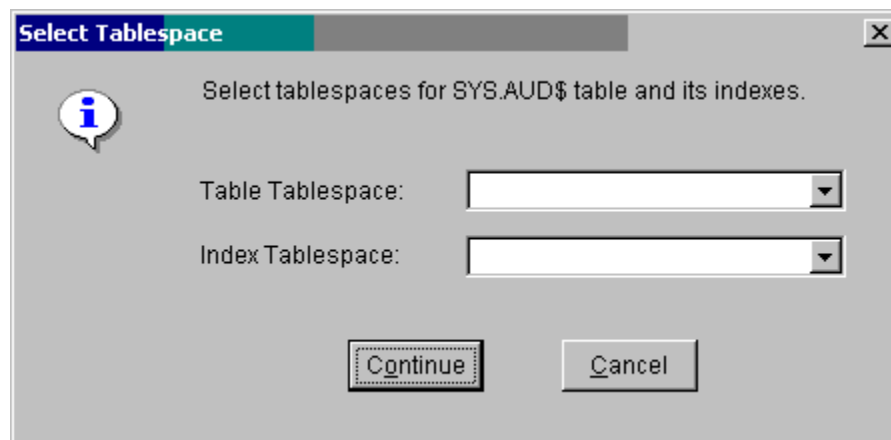
Oracle is somewhat unclear whether the SYS.AUD\$ table should be moved out of the SYSTEM tablespace. In the system administrators manual it says explicitly that Oracle does not recommend moving the SYS.AUD\$ table out of the SYSTEM tablespace because the Oracle code makes implicit assumptions about the data dictionary tables, such as SYS.AUD\$, which could cause problems with future upgrades. However in published Oracle DBA courses and technical support bulletins it recommends moving the table and event provides instructions on how to do that.

DB Audit provides a graphical interface for moving SYS.AUD\$ table with just a few mouse clicks. At any time later the table can be easily moved back to the SYSTEM tablespace if you wish so.



To move the SYS.AUD\$ table to a different tablespace:

1. Start DB Audit GUI console and connect to the database.
2. Click **Audit** menu then click **System Audit Trail** menu then click **Advanced Options** menu. The **Advanced Options** dialog will be displayed.
3. Click the **Move** button. The **Select the tablespace** dialog will appear.



4. Choose same or different tablespace for SYS.AUD\$ table and its indexes and then click the **Continue** button.

 **Notes:**

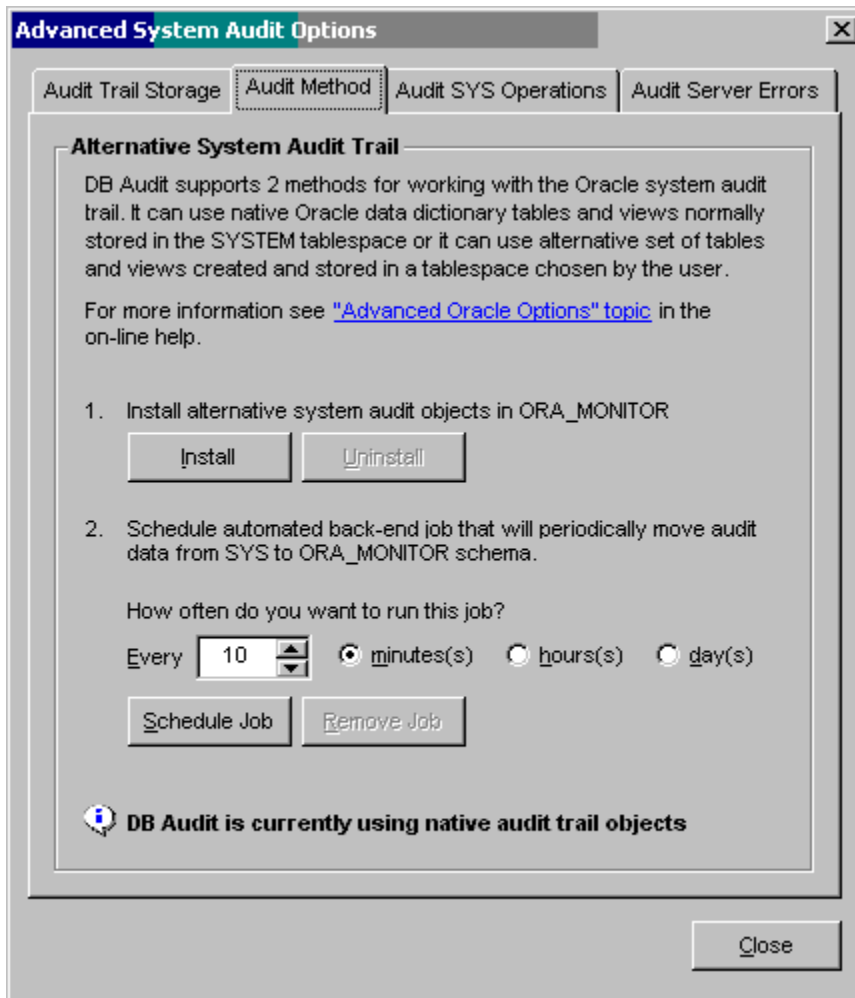
In Oracle 8.0 and later DB Audit uses ALTER TABLE and ALTER INDEX commands to move the table and indexes. In Oracle 7.x it creates a new AUD\$_NEW table, transfers the data, renames the current SYS.AUD\$ table to AUD\$_OLD and then renames AUD\$_NEW to AUD\$.

In all versions the existing system audit trail data is preserved.

It is important that you perform the move operation either when the auditing is turned off or nobody else is connected to the database.

Audit Method and Storage

DB Audit supports 2 methods for working with the Oracle system audit trail. It can use either native Oracle data dictionary tables and views (SYS.AUD\$ and DBA_AUDIT_...) normally stored in the SYSTEM tablespace or it can use alternative set of tables and views created in ORA_MONITOR schema and stored in a tablespace chosen by the user.



To install and switch to the alternative set of audit objects:

1. Start DB Audit GUI console and connect to the database.
2. Click **System Audit** menu then click **Advanced Options** menu. The **Advanced Options** dialog will be displayed.
3. Activate the **Audit Method** tab page.

4. Click the **Install** button. This will install all the required objects in the ORA_MONITOR schema. All objects are installed in the default tablespace chosen for ORA_MONITOR database user. In case if ORA_MONITOR schema and user don't exist yet DB Audit will prompt to select default tablespace and options and it will then create the required user.
5. Choose scheduling options for system audit data transfer job. This job will periodically move data from the system audit trail table SYS.AUD\$ to the alternative ORA_MONITOR.AUD\$ audit trail table.
6. Click the **Schedule** button to install the data transfer job

**Notes:**

Using alternative audit trail objects solves the space usage problem in the SYSTEM tablespace. However, this method may lead to high fragmentation of the SYSTEM tablespace if the amount of the generated audit data is quite large. For additional information about this issue read **Moving Oracle native system audit trail table out of the SYSTEM tablespace** section.

During the installation step your existing system audit trail data is automatically transferred to the alternative tables.

To uninstall the alternative objects click the Uninstall button and then remove the data transfer job. All existing audit trail data in the alternative tables will be lost when the objects are uninstalled.

Auditing Privileged Users connected as SYSDBA or SYSOPER

Although Oracle's audit facilities are very sophisticated they by default don't cover all user activities. For example, it is not possible to audit the actions of privileged users such as SYS and users connecting "as sysdba" until Oracle 9iR2.

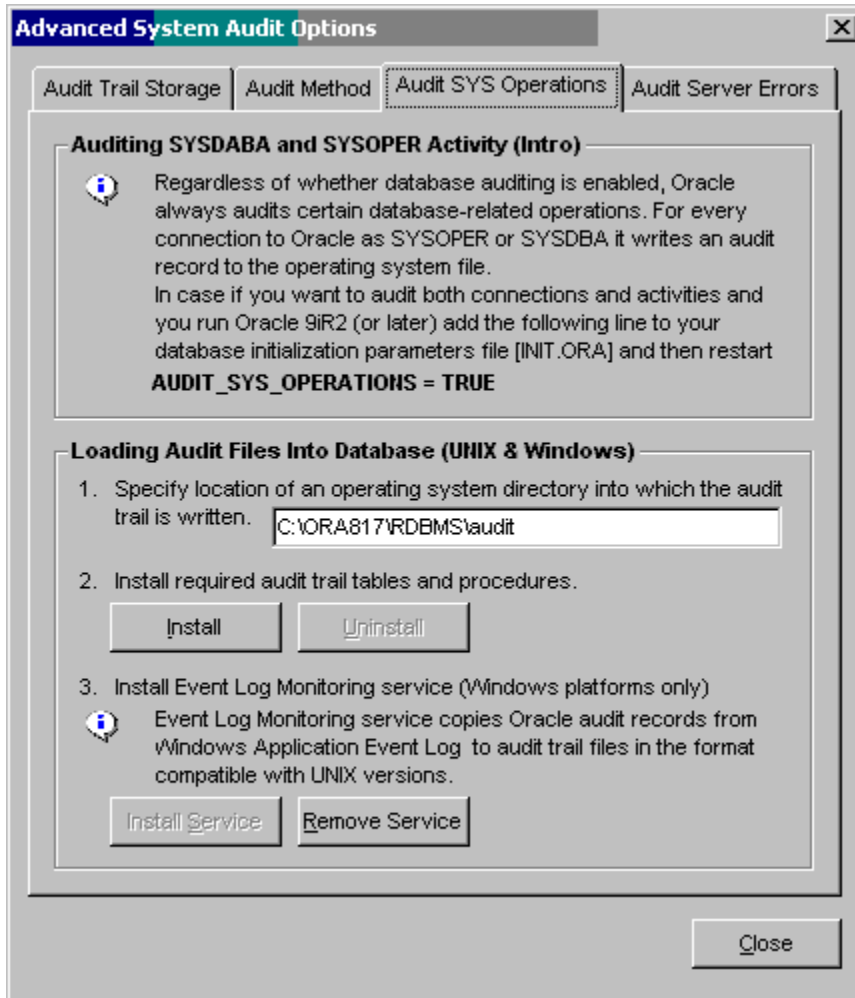
When a user with SYSDBA or SYSOPER privileges connects to the database, the action is expected to be for administrative reasons only, such as database shutdown, startup, configuration changes, etc. These actions are not intended for auditing and therefore the Oracle auditing mechanism does not pick them up.

Starting with Oracle release 9iR2, the database provides the full audit ability of the SYS account. In all versions prior to 9iR2 the only limited auditing option available is the small audit file that Oracle creates by default in the directory \$ORACLE_HOME/RDBMS/audit (in a Unix environment) or as a brief audit record written to the Windows NT application Event Log in Windows environments. This file is created every time a user attempts to connect internal through server manager (svrmgrl), or the now current method of connect as SYSDBA or SYSOPER. A separate file is written for every connection making it really hard to analyze the audit trail data and monitor security breaches.

In case if you want to audit both connections and activities and you run Oracle 9iR2 (or later) add the following line to your database initialization parameters file [INIT.ORA] and then restart the database.

AUDIT_SYS_OPERATIONS = TRUE

DB Audit provides methods for monitoring Oracle audit files and loading them into a database table. This somewhat fulfills the gap for Oracle database version prior to version 9iR2. The data is loaded into ORA_MONITOR.SYSFILE_AUD\$ table. This table can be queried using DB Tools' SQL Editor or any other reporting tool of your choice. You can also use the Report Designer tool available in DB Monitor Expert (see Custom Reports topic) to design a report that you can run later at any time with a click of a mouse.



To install audit of SYS-type operations:

1. Start DB Audit GUI console and connect to the database.
2. Click **Audit** menu then click **System Audit Trail** menu and then click **Advanced Options** menu. The **Advanced Options** dialog will be displayed.
3. Activate the **Audit SYS Operations** tab page.
4. Specify location of an operating system directory into which the audit trail is written. You must specify absolute directory name and cannot use system environment variables pointing to Oracle home. Do not use trailing slash in the directory name.

 **Notes:**

In Unix environments the directory name is case-sensitive. Make sure to enter it exactly.

Important: The specified audit directory must be listed in the initialization parameter `UTL_FILE_DIR`. Please add it to your database initialization parameters file [INIT.ORA], restart the database and then install Audit SYS Operations procedures. Here is an example line from [INIT.ORA] file:


```
util_file_dir = c:\oracle\mydb\rdbms\audit
```

In case if you already have `UTIL_FILE_DIR` parameter in your [INIT.ORA] parameters file referencing some other directory you can add more lines. To specify multiple directories have

a separate UTL_FILE_DIR line for each directory, but make sure these are on consecutive lines otherwise only the last directory will be accessible.

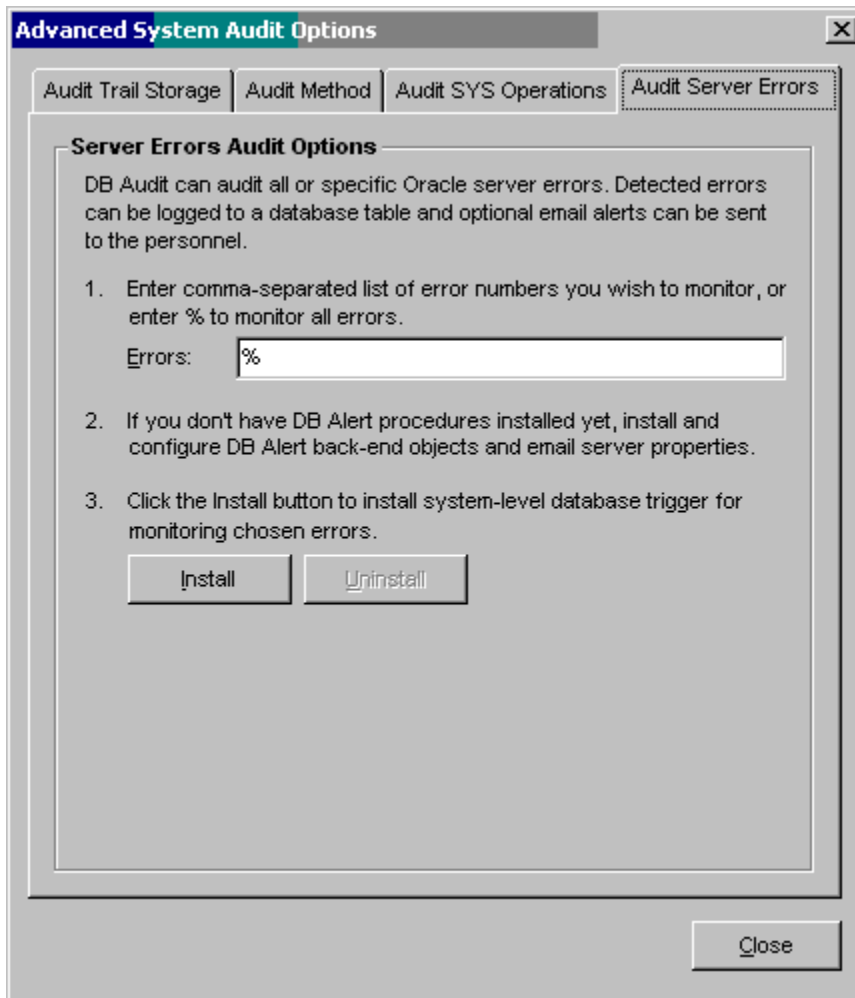
5. Click the **Install** button. This will install all the required objects in the ORA_MONITOR schema. All objects are installed in the default tablespace chosen for ORA_MONITOR database user. In case if ORA_MONITOR schema and user don't exist yet DB Audit will prompt to select default tablespace and options and it will then create the required user.
6. If your Oracle database runs on a Windows server install DB Audit Event Log Monitoring service. The Event Log Monitoring service copies Oracle audit records from the Windows Application Event Log to audit trail files in the format compatible with Unix versions.

 **Important:** The service must be installed on the computer running your Oracle server, not the computer running DB Audit GUI. You must install DB Audit console on the server computer in order to install the Event Log Monitoring service.

 **Note:** The audit trail data is loaded into ORA_MONITOR.SYSFILE_AUD\$ table. This table can be queried using DB Tools' SQL Editor or any other reporting tool of your choice. You can also use the Report Designer tool available in DB Monitor Expert (see Custom Reports topic) to design a report that you can run later at any time with a click of a mouse.

Auditing Oracle Database Errors

In Oracle versions 8i and later DB Audit can take advantage of Oracle system-level triggers to audit Oracle server errors. DB Audit can audit all or specific Oracle server errors. Detected errors can be logged to a database table and optionally email alerts can be sent to the personnel.



To install procedures for automatic capturing of database errors:

1. Make sure [DB Alert Expert](#) procedures are installed in the database.
2. Start DB Audit GUI console and connect to the database.
3. Click **Audit** menu then click **System Audit** menu then click **Advanced Options** menu. The **Advanced Options** dialog will be displayed.
4. Activate the **Audit Server Error** tab page.
5. In case if you want to audit specific Oracle errors enter Oracle error numbers as comma-separated list, otherwise enter % (percent sign) to capture and audit all errors.
6. If you want to implement email alerts specify email alert recipients, if none is specified no email will be sent.



Important: The DB Alert procedure and console must be installed and configured prior to installing database error auditing. For more information see [Configuring DB Alert](#) topic.

7. Click the **Install** button. This will install all the required objects in the ORA_MONITOR schema. All objects are installed in the default tablespace chosen for ORA_MONITOR database user. In case if ORA_MONITOR schema and user don't exist yet DB Audit will prompt to select default tablespace and options and it will then create the required user.



Note:

The audit trail data for database errors is loaded into ORA_MONITOR.SERVER_ERROS table. This table can be queried using DB Tools' SQL Editor or any other reporting tool of your

choice. You can also use the Report Designer tool available in DB Monitor Expert (see Custom Reports topic) to design a report that you can run later at any time with a click of a mouse.

Data Change Auditing

DB Audit Expert enables the tracking of data changes made to a database. DB Audit Expert can record what changes were made and who, when, and from where they were made. Any number of tables or columns may be selected for auditing. Different types of auditing can be specified for different tables. For instance, you can track only inserts on one table and monitor all kinds of changes on another table.

As indicated before, the actual auditing of the database is implemented by installing audit triggers on each table selected for auditing and also creating audit trail tables to store data from the changed records. The installed data-change auditing is completely transparent to both the existing as well as new applications. All supported database systems allow multiple triggers defined on a table. Hence, even if you already have some existing application triggers or add new triggers later both the application and audit triggers will work simultaneously. In short, the audit triggers will not impact or affect existing triggers.

For each audited table DB Audit generates an "audit" trigger and a "mirror" table the ORA_MONITOR schema. If this schema does not exist, DB Audit Expert asks your permissions to create it whenever you select the very first table to be audited. DB Audit Expert generates both the "audit" trigger and the "mirror" table automatically. The "mirror" table is used as a data audit trail table for storing data changed in the audited table. This history table stores one auditing record for every deleted or newly inserted record and two records for every updated record. In case of updates, the first record always stores old value and second record stores updated value.

The data audit trail table has the same structure as the audited table. It also includes columns to store additional information about made changes such information as the user name who changed the data, terminal from where the "change" operation was initiated and the timestamp that indicates exact time of the change.

If the table being audited includes at least one unique column, then you can also use data from the data audit trail to recover the audited table in cases of data loss.

Note that DB Audit Expert is capable to generate multiple triggers on multiple selected tables simultaneously, meaning that you can perform data change auditing on virtually unlimited number of tables simultaneously

Because volume of audit trail information can grow very fast you should regularly archive audit records and purge the audit trail.

Once you have collected the required information, archive the audit records of interest and purge the audit trail.

Before You Begin Data Auditing

You must have SELECT ANY TABLE, CREATE TABLE, CREATE TRIGGER, REFERENCE ANY privileges.

Privileges required

The table below describes default privileges that must be granted to the ORA_MONITOR user.

Oracle 7.3, 8.0 and 8i

```
SELECT ANY TABLE
EXECUTE ANY PROCEDURE
```

Oracle 9i and 10g requires explicit grants given by SYS

```
SELECT ON sys.v_$session
SELECT ON sys.v_$mystat
EXECUTE ON sys.dbms_lob
EXECUTE ON sys.dbms_output
EXECUTE ON sys.dbms_job
EXECUTE ON sys.dbms_util
SELECT, DELETE ON sys.audit$
```

Archiving data audit trail to a table

1. Select **File/Data Audit Trail/Archive to Table** command from the DB Audit Expert menu.
2. The select source and destination table dialog appears. Select name of the existing audited table from the source table list.
3. Specify either name of an existing table to which you archived before using DB Audit Expert or enter the name of a new table. DB Audit Expert automatically creates the destination table if it does not exist. Note that if have not purged the data audit trail since last archiving operation, you will get the same audit trail records archived twice.
4. If necessary, repeat described steps for another table being audited.

Archiving data trail to a file

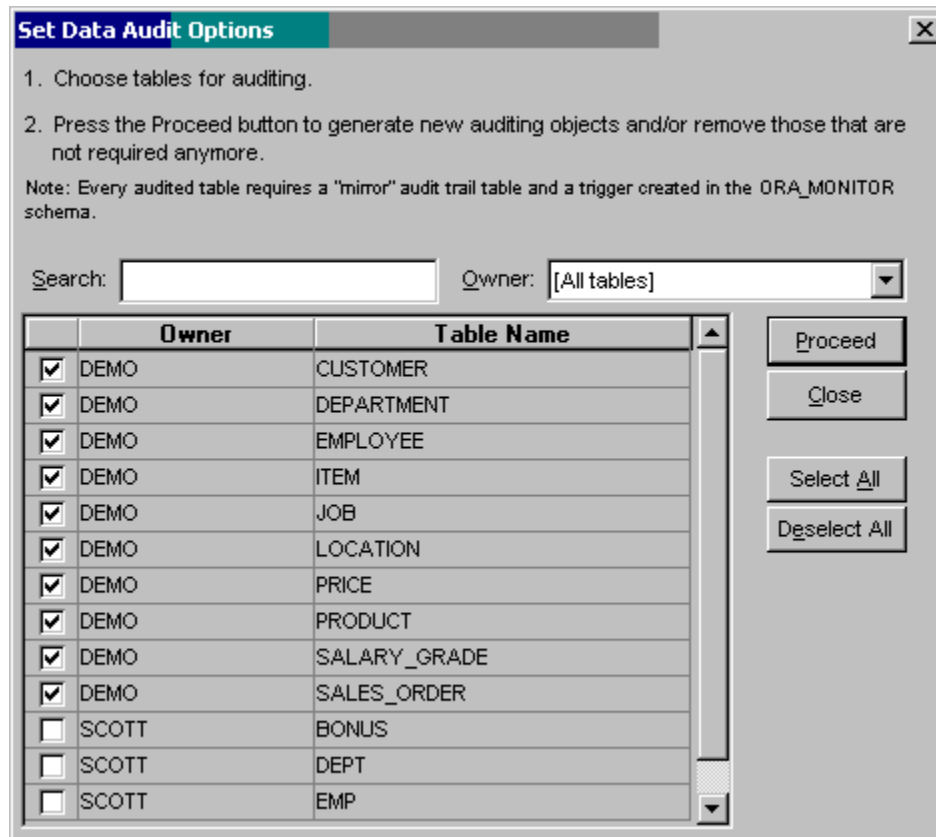
1. Select **File/Data Audit Trail/Export to File** command from the DB Audit Expert menu.
2. The select export file dialog appears. Specify name of the file to which you want to archive the data audit trail. If you select an existing file, DB Audit Expert will overwrite this file.
3. The select table dialog appears. Select name of the table for which you want to archive the data audit trail.
4. If necessary, repeat described steps for another table being audited.

Purging data audit trail:

1. Select **File/Data Audit Trail/Purge** command from the DB Audit Expert menu.
2. The select audited table dialog appears. Select name of the table for which you want to purge the data audit trail.
3. If necessary, repeat described steps for another table being audited.

Enabling Data Change Auditing

1. To select tables for auditing, use **Audit/Data Audit Trail/Generate| Remove Triggers** command from the DB Audit Expert menu. The Set Data Audit options dialog will appear.



2. Check all tables that you want to audit. When ready click the **Proceed** button.

Disabling Data Change Auditing

1. To remove previously configured audit triggers use **Audit/Data Audit Trail/Generate| Remove Triggers** command from the DB Audit Expert menu.
2. The Set Data Audit options dialog will appear.
3. Uncheck all tables that you don't want to audit anymore. When ready click the **Proceed** button.

DB Patch Expert

Overview

DB Patch Expert provides graphical interface and reporting tools for the web-based Patch Monitoring and Management Service. The Patch Monitoring and Management Service is responsible for monitoring and collecting information about new database patches and upgrades. It can also help you with prioritization and scheduling of your database updates, ensuring that patches with critical security and functionality fixes are installed in a timely fashion. The Patch Monitoring and Management Service can be optionally configured to handle general patch management tasks such as keeping track of patch testing and installation history, correlating new security threats and patch availability, as well as performing other important patch management tasks.

 **Note about the Patch Monitoring and Management Service:** SoftTree Technologies, Inc runs one of the largest archives of database patch information. The archive is self-updating and always up to date. SoftTree servers perform automated monitoring of new patch releases for all major database types, including Oracle, SQL Server, Sybase, DB2 and MySQL and automatically add them to the archive.

For licensed users, the Patch Monitoring and Management Service provides additional services such as storing and managing history of patch installations and system upgrades, automatic alerting and reporting on new patch releases, analytical and compliance reporting.

The Patch Monitoring and Management Service requires each user to register with the service. All basic services are free for 30-days for all registered users. After the initial 30-day period, users are required to subscribe to the service and renew their registration annually in order to continue using the service and continue receiving new patch notifications. Users can unsubscribe at any time or simply delete registration records associated with their database systems.

Patch Analysis and Reporting

Patch Summary Report

This reports analyzes your current database settings and display summary statistic for patches released and applicable to your database system. It reports totals numbers of all available patches, patches that are already installed on the system or included in your version, and patches that could be installed with a summary breakdown by patch severity. Summary report for an Oracle server version 9.0.1.1 running on Windows system may look like the following example:

Summary Statistics

for Oracle v9.0.1.1 for Windows 32-bit platform

Statistic	Total
Number of patches and upgrades since release 9.0.1.1 (including new major versions)	3229
Installed patches and upgrades for release 9	61
New patches and upgrades (NOT including new major versions)	1280

*Patches and Upgrades Released**Since Oracle v9.0.1.1 for Windows 32-bit platform*

Patch Severity	Total
Critical	9
High	0
Medium	1271
Low	0
Unknown and not classified	0
Grand Total	1280

To access this report, use **Patches** menu in the DB Patch Expert.

Installed Patches Report

This reports analyzes your current database settings and display detailed information about patches that are already installed on the system or included in your version. The displayed information includes patch issue date, type, platform, dependencies, summary information and installation details and instructions, as well as the path download link.

To access this report, use **Patches** menu in the DB Patch Expert.

Missing Critical Patches Report

This reports analyzes your current database settings and display detailed information about patches with known critical severity that should be installed on your Oracle server, but not yet installed. The displayed information includes patch issue date, type, platform, dependencies, summary information and installation details and instructions, as well as the path download link.

To access this report, use **Patches** menu in the DB Patch Expert.

Missing Critical and High Severity Patches Report

This reports analyzes your current database settings and display detailed information about patches with known critical and high severity that should be installed on your Oracle server, but not yet installed. The displayed information includes patch issue date, type, platform, dependencies, summary information and installation details and instructions, as well as the path download link.

Note that "Missing Critical Patches Report" report returns a subset of records returned by this report.

To access this report, use **Patches** menu in the DB Patch Expert.

Missing Critical, High, and Medium Severity Patches

This reports analyzes your current database settings and display detailed information about patches with known critical, high, and medium severity that are available for your Oracle server, but not yet installed. The displayed information includes patch issue date, type, platform, dependencies, summary information and installation details and instructions, as well as the path download link.

Note that "Missing Critical and High Severity Patches Report" report returns a subset of records returned by this report.

To access this report, use **Patches** menu in the DB Patch Expert.

All Missing Patches

This reports analyzes your current database settings and display detailed information about all patches available for your Oracle server, but not yet installed. The displayed information includes patch issue date, type, platform, dependencies, summary information and installation details and instructions, as well as the path download link.


Note that "Missing Critical, High, and Medium Severity Patches Report" report returns a subset of records returned by this report.

To access this report, use **Patches** menu in the DB Patch Expert.


Automated Alerts and Notifications

The Patch Monitoring and Management Service can automatically notify you whenever new patches and upgrades are released for your database systems. In order to use this feature you first configure the service and enter versions and types of databases for which you want to get notifications and configure when and how you want to be notified. At the time of DB Tools 6.0 release, the Patch Monitoring and Management Service supported the following 3 notification frequencies and formats:

- **Instant notification** – this type of notification enables the service to send you email alerts for each database system you registered with the service as soon as new patches and upgrades become available for that system. The email alerts include patch details including download links and installation instructions (if available). Typically, such email alerts are sent next day after new patches become [publicly available].
- **Weekly notification** – this type of notification enables the service to send you consolidated email reports for all patches that become available during the previous week for all database systems you registered with the service. The report includes basic patch information and reference links to the patch service web site where you can find more information about referenced patches. Note that the report is not sent if no new patches and upgrades are released during the previous week.
- **Monthly notification** – this type of notification enables the service to send you consolidated email reports for all patches that become available during the previous month for all database systems you registered with the service. The report includes basic patch information and reference links to the patch service web site where you can find more information about referenced patches. Note that the report is not sent if no new patches and upgrades are released during the previous month.

 **Important Notes** On one hand, weekly and monthly notifications provide more convenience to users running multiple database servers. On the other hand, these reports are sent only periodically and they may come too late, leaving your database systems vulnerable to new security threats. If you are not sure why this is important, read in [Wikipedia](#) – the free on-line encyclopedia about the Slammer virus, which has spread very rapidly infecting tens of thousands of database servers in the first day alone.

To subscribe to automated patch notifications, use **Patches/New Patch Alerts** menu in DB Patch Expert. You would need to use that menu for database system for which you want to receive the notifications. The data is pre-filled automatically by DB Patch Expert. You should verify this data for accuracy, enter the email address of notification recipient and click the Submit button.

 **Note:** To have notifications sent to a group of people, for example, to all DBAs in your team, enter email address of your team email distribution group.


Patch Management Tasks

The Patch Monitoring and Management Service can help you to manage patch prioritization, scheduling, testing and deployment processes. It can keep track of installed patches and related activities. It also allows you use to analyze patch issuance and deployment history and improve the database change management process which is vital to every stage of the patch management.


Use **Patches/Patch and Upgrade Installation Tracker** menu in DB Patch Expert to access tools available for the patch history recording and analysis.

Note that when you access the tracker, it automatically reconciles the data stored in the Patch Management service database and versions of your database and in case of discrepancies it automatically adds new records to the patch installation log.


To add a new record to the patch tracker:

1. Connect to the database system whose records you want to view or modify.
2. Click **Patches/Patch and Upgrade Installation Tracker** menu
3. Click  icon or click **Add new record** hyperlink.
4. Fill in available fields in the new record following on the screen instructions.
5. Click the **Save** button.

To delete an existing record:

1. Connect to the database system whose records you want to view or modify.
2. Click **Patches/Patch and Upgrade Installation Tracker** menu.
3. Click  icon displayed in the left most column of the record you want to delete.
4. When prompted to confirm the operation, click the **Yes** button.

To modify an existing record to the patch tracker:

1. Connect to the database system whose records you want to view or modify.
2. Click **Patches/Patch and Upgrade Installation Tracker** menu
3. Click  icon displayed in the left most column of the record you want to modify.
4. Modify this record data as required.
5. Click the **Save** button.

DB Job Expert

Overview

DB Job Expert contains all tools required for creating database jobs, scheduling, launching, and monitoring, reporting and troubleshooting of jobs.

The graphical interface of DB Job Expert provides an enterprise view for quick, clear inspections and maintenance of all scheduled jobs, their analysis and troubleshooting. The back-end job monitoring and profiling package instantly monitors running jobs and collects their performance statistics. The collected statistics are used in a variety of DB Job Expert reports that vary by functionality from Job Forecasting to Server Load Balancing.

The DB Job Expert also includes Job Monitor package and a set of jobs that must be installed in the database in order to use advanced DB Job Expert features such as Job Profiler and job performance reports. The Job Monitor package is used to monitor scheduled jobs and collect job performance statistics. The Job Profiler uses collected job performance data in most DB Job Expert Reports. The collected data can be also used as a complete job audit log.

DB Job Expert is built around Oracle scheduling API provided in the standard DBMS_JOB package. In certain cases, DB Job Expert can use functions from the system undocumented DBMS_IJOB package. These functions allow SYS user to modify, disable, drop and run jobs that belong to other database users.

Before you start using scheduling functions, make sure you have properly configured your Oracle job queues and SNP background processes that actually run jobs. You must have at least one SNP process running to execute your scheduled and queued jobs. Note that SNP processes will not execute job queues if the database is in restricted mode. An instance can have up to 36 SNP background processes.

In order to use job queues you will need to set the following two INIT.ORA parameters:

1. JOB_QUEUE_PROCESSES = n

Here n is how many SNP background processes you want to be running.

Default: 10

Valid range of values: 0...36

2. JOB_QUEUE_INTERVAL = n

Here n is the interval between wake-ups, in seconds, for the SNP background processes.

Default: 60

Valid Range of values: 0...3600

Please be aware that if either JOB_QUEUE_PROCESSES = 0 or JOB_QUEUE_INTERVAL = 0, scheduled jobs will not run

See the following topics for more information on DB Job Expert

[Job Scheduler Console](#)

[Job Monitor Installation](#)

[Job Performance Statistics and Job Monitor Options](#)

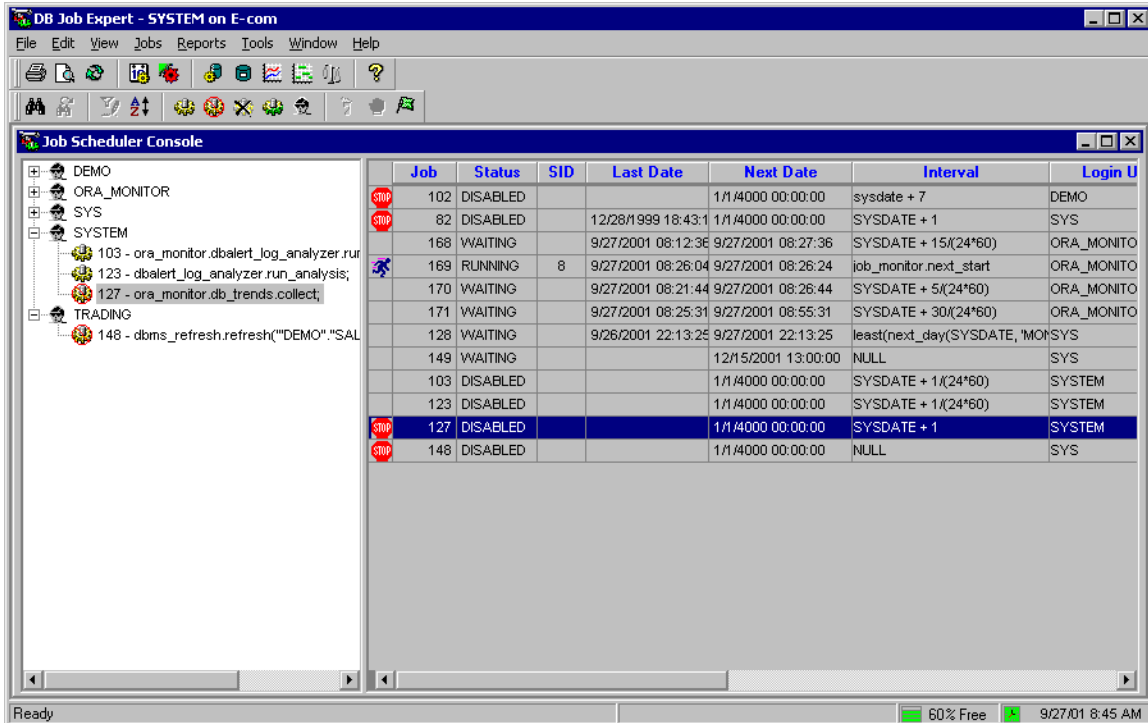
[Job Profiler](#)

[Job Reports](#)

Job Scheduler Console

The Job Scheduler Console is a visual configuration tool you can use to inspect already scheduled jobs, create new and modify/delete existing jobs, start jobs manually and forcibly terminate unrequired jobs

The Job Scheduler Console is automatically started when DB Job Expert GUI is started.



Navigation techniques


The Job Scheduling Console allows two different ways of viewing and modifying currently scheduled jobs:


- The tree-view on the left-hand of the console window presents jobs grouped by job owners.
- The grid-view displays all job details including job id, last job run-time, next job run-time recursion interval, job schema, job privileges and job SQL.

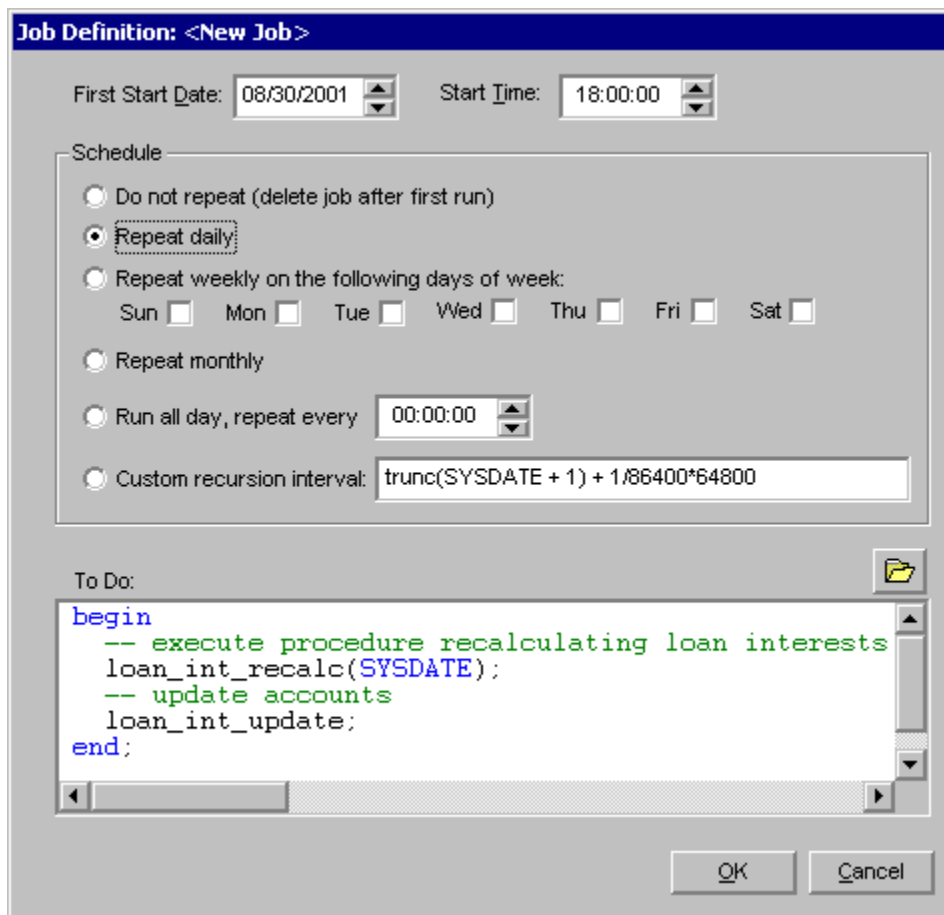
You may access controls for a job through the main menus, at the top of the Job Scheduler Console window; through the toolbar, which allows you to manipulate a selected job in the tree-view; or through the context-sensitive menu that appears when you right-click a job item in the tree-view or in the grid-view. All methods allow you to choose whether to edit, change, delete, run or stop a job.

Remember that while the main menu contains almost all controls and functions available in DB Job Expert, only the context-sensitive menus allow performing context operations such as Print and Print Preview. In other words, if you right-click the tree-view and then select the **Print** command from the popup context menu, this will print the contents of the tree-view, but If you right-click the grid-view and then select the **Print** command from the popup context menu, this will print the contents of the grid view.

Creating a new job

1. Click **Jobs/New** menu or click  toolbar button. The <New Job> Properties dialog will appear (see sample image below)
2. Choose job first start date and time

- Choose required job schedule. You can select one of the predefined common job schedules or enter your own job recursion formula into **Custom recursion interval** field. The formula must calculate the next time to run the job. The returned value must evaluate to either a future point in time or NULL. In this formula you can use standard Oracle date functions or your own SQL functions (Oracle 8i and later).
- Enter SQL statements to be executed, you can type them in the **To Do** box or use the Browse  button to paste text from a file. The SQL in To Do box must be either a single valid SQL statement such as name of stored procedure or an Oracle anonymous PL/SQL block beginning with DECLARE or BEGIN keyword and containing one or more valid PL/SQL statements.
- Click the OK button to create the new job.



Job Definition: <New Job>

First Start Date: 08/30/2001 Start Time: 18:00:00

Schedule

Do not repeat (delete job after first run)

Repeat daily

Repeat weekly on the following days of week:
 Sun Mon Tue Wed Thu Fri Sat

Repeat monthly

Run all day, repeat every 00:00:00



Custom recursion interval: trunc(SYSDATE + 1) + 1/86400*64800

To Do:

```
begin
  -- execute procedure recalculating loan interests
  loan_int_recalc(SYSDATE);
  -- update accounts
  loan_int_update;
end;
```


OK Cancel

Modifying properties of an existing job


- Select the required job in the tree-view or grid-view
- Click **Jobs/Modify** menu or click  toolbar button. The Modify Job Properties dialog will appear
- If necessary, change job next start date and time
- If necessary, change job schedule. You can select one of the predefined common job schedules or enter your own job recursion formula into **Custom recursion interval** field. The formula must calculate the next time to run the job. The returned value must evaluate to either a future point in time or NULL. In this formula you can use standard Oracle date functions or your own SQL functions (Oracle 8i and later).
- If necessary, change SQL to be executed next time the job runs, you can type them in the **To Do** box or use the Browse  button to paste text from a file. The SQL in To Do box must be either


a single valid SQL statement such as name of stored procedure or an Oracle anonymous PL/SQL block beginning with DECLARE or BEGIN keyword and containing one or more valid PL/SQL statements.


6. Click the OK button to modify the new job.

 **Important Note:** You can modify your own jobs without any restrictions, however you must use SYS logon if you need to modify a job that was created by another user.


Toggling job disabled/enable state


1. Select the required job in the tree-view or grid-view
2. Click **Jobs/Disable/Enable** menu or click the  toolbar button. The job state will be toggled. If the job new state is disabled the job will never run until it is re-enabled.

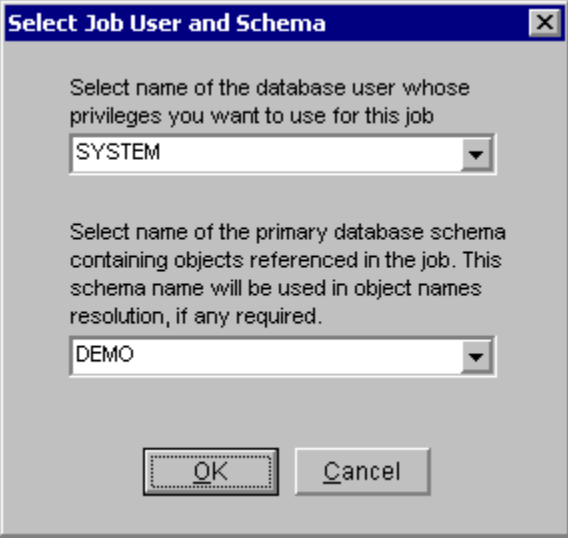
 **Tip:** Disabled jobs can run only if they are invoked manually or from other jobs.

 **Important Note:** You can disable and enable your own jobs without any restrictions, however you must use SYS logon if you need to disable or re-enable a job that was created by another user.

Changing job privileges

1. Select the required job in the tree-view or grid-view
2. Click **Jobs/Change Job Privileges** menu or click  toolbar button. The job permissions dialog will appear (see sample image below)
3. Choose name of the user whose privileges are required to run the job.
4. Choose name of the user who owns the objects specified in the job SQL (see **To Do** properties in the Modify Job subtopic)
5. Click the OK button.

 **Important Note:** You must use SYS logon in order to be able to change job privileges.



Select Job User and Schema

Select name of the database user whose privileges you want to use for this job


SYSTEM


Select name of the primary database schema containing objects referenced in the job. This schema name will be used in object names resolution, if any required.


DEMO

OK Cancel


Removing an existing job

1. Select the required job in the tree-view or grid-view
2. Click **Jobs/Drop** menu or click the  toolbar button. The confirmation dialog will appear and the job will be removed if you confirm the operation.

 **Tip:** Removing job definition from the job queue does not stop a running job. To kill a running job use **Jobs/Kill Job** command. For more details see the Kill Running Job subtopic below.


 **Important Note:** You can remove your own jobs without any restrictions, however you must use SYS logon if you need to remove a job that was created by another user.

Running a job on demand


1. Select the required job in the tree-view or grid-view
2. Click **Jobs/Run Now** menu or click the  toolbar button. The job will begin executing in the foreground and the Job Status window will appear.


 **Tips:**

- Remember that a manual job run may affect job schedule if the job recursion interval is defined as a straight function of SYSDATE, for example SYSDATE + 1.
- To interrupt the manually started job, click the Kill Session button on the Job Status window, this will kill the DB Job session running the job. You will need to reconnect DB Job Expert to the database in order to continue working with jobs.
- To run the job in background, copy job definition to a new job and choose SYSDATE for the first start time and NULL for the recursion interval.


 **Important Note:** You can run your own jobs without any restrictions, however you must use SYS or other administrative logon if you need to run a job that was created by another user.


Stopping a running job

1. Select the required job in the tree-view or grid-view
2. Click **Jobs/Kill Job** menu or click the  toolbar button. This will kill the session that is associated with the job and the job will stop.

 **Important Note:** You must have sufficient privileges to terminate database sessions.

Retrieving job log and statistics

1. Select the required job in the tree-view or grid-view
2. Click **Jobs/Profiler Statistics** menu or click the  toolbar button. The Job Profiler Report will be displayed for the selected job.

 **Important Note:** The DB Tools Job Monitor package must be installed and running on the database in order to be able to get job logs and performance statistics. For more information see [Job Monitor Installation](#) topic.

Job Monitor Installation

To install Job Monitor use the graphical DB Tools [Database Setup utility](#). Follow instructions provided by the Setup Wizard.

DB Job Expert is built around Oracle scheduling API provided in the standard DBMS_JOB package. In certain cases, DB Job Expert can use functions from the system undocumented DBMS_IJOB package. These functions allow SYS user to modify, disable, drop and run jobs that belong to other database users.

Before you start using scheduling functions, make sure you have properly configured your Oracle job queues and SNP background processes that actually run jobs. You must have at least one SNP process running to execute your scheduled and queued jobs. Note that SNP processes will not execute job queues if the database is in restricted mode. An instance can have up to 36 SNP background processes.

In order to use job queues you will need to set the following two INIT.ORA parameters:

1. JOB_QUEUE_PROCESSES = n

Where n is how many SNP background processes you want running.

Default:

Range of values: 0...36

2. JOB_QUEUE_INTERVAL = n

Where n is the interval between wake-ups, in seconds, for the SNP background processes.

Default: 60

Range of values: 0...3600

Please be aware that if either JOB_QUEUE_PROCESSES = 0 or JOB_QUEUE_INTERVAL = 0, scheduled jobs will not run

See also:

[Job Performance Statistics and Job Monitor Options](#)

Job Performance Statistics and Job Monitor Options

Job Performance Statistics

Job Monitor collects statistics for every running job. It optionally collects overall system statistics during job runs and during periods of time when no job is running. The job statistics are collected in the following events:

- Immediately after job start
- One or more times during job run (the actual number of collection and their intervals depend on the number of other simultaneously running jobs).
- Immediately after job finish

The collected job statistics are used in a variety of DB Job Expert reports. Collected system statistics can be used in custom build reports (you can use the Performance Library tools) to find windows in server heavy processing so you can better balance database load created by jobs and other processes.

All collected statistics are written into the ORA_MONITOR.JOB_STAT table.

Not all statistics are available by default. Statistics such as **CPU usage** are available only when Oracle **TIMED_STATISTICS** option is turned on. If you want to collect CPU and other timed statistics you must turn on **TIMED_STATISTICS** option in INIT.ORA or each time the system starts turn them on using the following command


```
ALTER SYSTEM SET timed_statistics=TRUE;
```

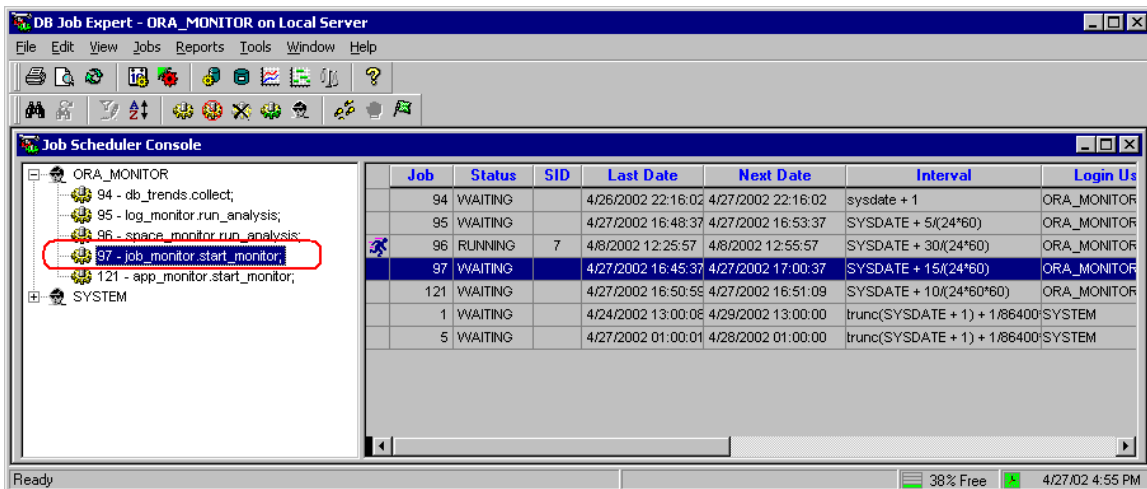
Job Monitor Options

To change Job Monitor parameters use Tools/Job Monitor Options menu in the DB Job Expert GUI

The following parameters can be changed.

1. Job Check Interval -- This parameter defines job status check interval in seconds. High values may result in poor statistics accuracy but lead to less CPU consumption. Cost of job status check operation is roughly equivalent to the cost of `select * from v$lock where type = 'JQ'` query. Recommended value is half of the average job duration.
2. Max Record Interval -- This parameter defines in seconds maximum period of time to collect and record all jobs & system statistics to the repository table JOB_STAT. It is also maximum period of time for the monitor job to be inactive in the job queue. High values result in lazy detection of new jobs. Low values result in extra CPU usage and increased size of the repository table.
3. Record System Statistics (Yes/No) -- This parameter controls whether the job monitor should also record system statistics between and during job runs. This information can be used in finding windows for job runs and other job performance analyses.

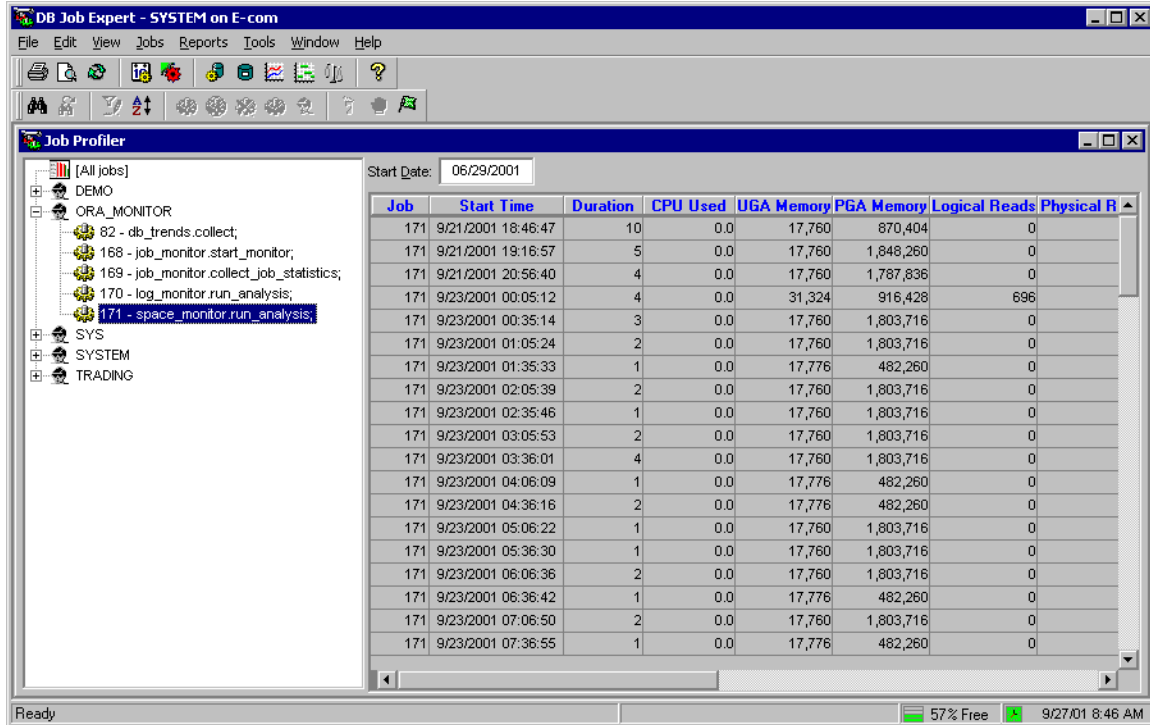
 **Important Note:** The statistics collected by the Job Monitor are as accurate as accurate are settings for the Job Monitor job (see `job_monitor.start_monitor` job on the following picture). If the Job Monitor settings are incorrect it may lead to a situation when the Job Monitor will miss important jobs.



Job	Status	SID	Last Date	Next Date	Interval	Login Us
94	WAITING		4/26/2002 22:16:02	4/27/2002 22:16:02	sysdate + 1	ORA_MONITOR
95	WAITING		4/27/2002 16:48:37	4/27/2002 16:53:37	SYSDATE + 5/(24*60)	ORA_MONITOR
96	RUNNING	7	4/8/2002 12:25:57	4/8/2002 12:55:57	SYSDATE + 30/(24*60)	ORA_MONITOR
97	WAITING		4/27/2002 16:45:37	4/27/2002 17:00:37	SYSDATE + 15/(24*60)	ORA_MONITOR
121	WAITING		4/27/2002 16:50:55	4/27/2002 16:51:09	SYSDATE + 10/(24*60*60)	ORA_MONITOR
1	WAITING		4/24/2002 13:00:06	4/29/2002 13:00:00	trunc(SYSDATE + 1) + 1/86400	SYSTEM
5	WAITING		4/27/2002 01:00:01	4/28/2002 01:00:00	trunc(SYSDATE + 1) + 1/86400	SYSTEM

Job Profiler

The Job Profiler is the tool within DB Job Expert that is capable to retrieve, display and analyze job performance statistics collected by the Job Monitor.



Navigation Techniques

The Job Profiler GUI consists of two parts:

- The tree-view on the left-hand of the Job Profiler window presents jobs grouped by job owners.
- The grid-view displays collected job performance statistics for a job if a job is selected in the tree-view or it displays complete statistics for all jobs if the very first item **[All jobs]** is selected in the tree-view.

If in your database you have many frequently running jobs, there could be a lot of the job performance data collected. To reduce the number of rows retrieved to the screen use the **Start Date** field displayed on the top of the Job Profiler window to set a filter for the output.

Remember that while the main menu contains almost all controls and functions available in DB Job Expert, only the context-sensitive menus allow performing context operations such as Print and Print Preview. In other words, if you right-click the tree-view and then select the **Print** command from the popup context menu, this will print the contents of the tree-view, but if you right-click the grid-view and then select the **Print** command from the popup context menu, this will print the contents of the grid view.

Job Reports

DB Job Expert currently supports 10 different job forecasting, job performance and job troubleshooting reports. Each report can be viewed and printed from the Report window. Most of these reports are based on the job performance statistics collected by the [Job Monitor package](#).

The following reports are available:

- [Job Forecast](#)
- [Job Resource Consumption](#)

[Job Performance & Trends](#)
[Job Timing and Conflicts](#)
[Database Load Balancing \(Job Distribution\)](#)
[Database Load Balancing \(CPU Load\)](#)
[Top 10 Job by CPU Usage](#)
[Top 10 Job by Memory Usage](#)
[Top 10 Job by Sorts](#)
[Top 10 Job by Disk I/O](#)

You can use them to forecast jobs, spot job trends, tune and troubleshoot jobs, and provide others with information through printing. Click on a report name in the list above for more information.

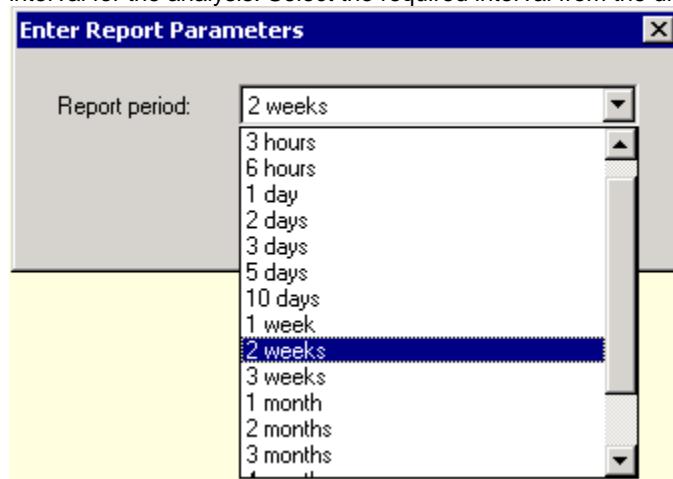


Tips:

- Most reports use job performance data for job processing analysis. This data is not available automatically. To allow job performance data collection you must install the back-end [Job Monitor](#) package and jobs. Do not run reports immediately after you install these objects. Wait as long as necessary for each job to run at least 4-5 times. This will allow the Job Monitor to gather sufficient volume of job performance data for an accurate job analysis. Do not force/trigger jobs manually for the purpose of performance data collection, as it will lead to inaccurate reports. The accuracy of job reports is proportional to the volume of gathered data. Longer you wait, more jobs run, and more valuable job performance data can be gathered.
- Job performance data is stored in the ORA_MONITOR.JOB_STAT table. If for some reasons you cannot find a pre-built job report that fits your needs, feel free to create your own reports using the [Performance Library](#) tools. For description and usage of JOB_STAT table see source code for the Job Monitor package provided with DB Tools.


Viewing reports:

- 1 In the Report menu, choose the type of report you want. The report window will appear.
- 2 If the selected report analyses historical job performance data, you will be prompted to choose time interval for the analysis. Select the required interval from the drop-down list and click OK.




- 3 You can use the Zoom feature to zoom in and out. You can either select one of the predefined values in the zoom drop-down box or type in the value you want then press the Enter key.

Printing reports:

- 1 Run report as described above.
- 2 When the report is ready, click the Print  toolbar button or select **File/Print** command from the main menu

Refreshing reports data:

- 1 Click the Refresh  toolbar button or select **View/Refresh** command from the main menu.
- 2 If the selected report analyses job performance data, you will be prompted to choose time interval for the analysis. Select the required interval from the drop-down list and click OK.

Job Forecast Report

This report calculates 7-day job forecast for all enabled job. This report can help you identify jobs you can expect to run during the forecast period.

Frequently running jobs with less than 15 minutes recursion interval are not shown on this report in order to improve the report's readability. You would know about these jobs anyway.

The report presents calculated data by day of week, sorted by job forecasted start time

Job Resource Consumption Report

This report is based on statistical analysis of job performance data collected by the Job Monitor.

This report shows trend chart for the total system load created by all background jobs that ran during the selected report period and also shows the system load created by individual jobs. You can use this report for the following purposes:

- 1 To identify jobs consuming lots of system resources.
- 2 To evenly distribute your system load between different jobs and applications so that the resource intensive applications do not run at the same time. This information can help you in improving the overall system performance, improving application response times and meeting required service levels.

The chart is scaled to fit the selected report period. You can try different report periods to see how the system is loaded over different periods of time. **Be aware that selecting longer report periods allows more job runs to be analyzed and thus it improves the report accuracy, but it can also increase the report calculation time.**

The following columns and values are displayed on the report
CPU Usage (chart) – Total CPU time spent for executing jobs

Time (chart) – System time

Job ID – Unique job ID in the database

Job SQL – The job To-Do

Schema – The name of the schema containing major job object

Total Executions – Number of times the job was executed during the selected report period

Average CPU Time – The average CPU time spent per job run during the selected report period

Total CPU Time - Total CPU time spent for all job runs during the selected report period



Tips:

It is a good idea to run [Job Timing and Conflicts](#) and [Server Load Balancing](#) reports that together with Job Resource Consumption Report will provide you with a complete picture of the system load and thus allow you to figure out how to use your system more efficiently.



Note:

Failed jobs and jobs whose execution took less than 1 second of CPU time are NOT included in this report

Job Performance and Trends Reports

This report calculates charts for top 20 jobs whose run-time duration have changed during the selected report period. This report is based on statistical analysis of job performance data collected by the Job Monitor.

You can use this report for the following purposes:

- 1 Identifying jobs whose run-time duration significantly changed from run-to-run during the chosen report period.
- 2 Identifying job duration trends (i.e., how quickly the duration changes over time) and thus you can use this information to forecast future durations.

You should choose a meaningful time interval for the report to allow the analyzed jobs to run for at least 4-5 times. Keep in mind that the trend-chart accuracy for each job highly depends on the number of successful job runs during the chosen report period.

Jobs with less than 3 executions during the chosen report period are ignored when the report is built.

The chart is scaled to fit the selected report period. You can try different report periods. **Be aware that selecting longer report periods allows more job runs to be analyzed and thus it improves the report accuracy, but it can also increase the report calculation time.**

The following values are displayed on the report

Duration – Job duration in seconds

Time – System time

Legend – Chart legend that shows up to 20 jobs presented on the chart. Each legend label contains both job ID and job SQL.

Colored solid lines on the chart indicate actual job durations over time. Colored dashed lines on the chart indicate job duration trends. Job trend lines are built using simple linear regression model. Both actual and trend lines are drawn using the same unique color assigned to each job.

Job Timing and Conflicts Report

This report calculates Gantt chart for all jobs that ran during the selected report period. This report is based on statistical analysis of job performance data collected by the Job Monitor.

You can use this report for the following purposes:

- 1 To identify concurrent or conflicting jobs, those that happen to run at the same time or delay other jobs. This information can help you in job troubleshooting.
- 2 To determine time windows when the system is not busy executing jobs or executing too many jobs simultaneously. You can use this information for the system load balancing.
- 3 This report also complements [Job Resource Consumption](#) report and both reports can be used to identify jobs that cause high system load. However, keep in mind that the Job Timing and Conflicts report charts job executing times, and NOT job CPU usage time. Long running jobs do not always take a lot of system resources.

The chart is scaled to fit the selected report period. You can try different report periods. **Be aware that large report periods may prevent you from seeing jobs whose duration is very small as compared to the chosen report period. If the chart appears empty try selecting smaller report period.**

The following values are displayed on the report

Time – System time

Job ID – Unique job ID in the database

Legend – The chart legend describing the selected database server. Future versions of this report may display side-by-side statistics from different servers.

Database Load Balancing Report (Job Distribution)

This report is based on statistical analysis of job performance data collected by the Job Monitor. This report can be helpful if you have numerous database jobs running on your server. The Database Load Balancing Report can help you identify time windows with low server utilization and schedule new or modify existing jobs to run during these times.

The report contains two types of load charts: the "load by day of week" chart and the "load by time of day" chart. Both charts display database utilization as a percent of all jobs executed during the chosen report period.

Each chart is calculated using job performance data for the chosen report period. You can try different report periods. **Be aware that selecting longer report periods allows more job runs to be analyzed and thus it improves the report accuracy, but it can also increase the report calculation time.**

The following values are displayed on the report

Time – Time of day (1 day scale, 24 hours)

% of All Jobs – The relative number of concurrent jobs running at a given point of time. This is calculated as a percent value using the following formula:

1 – number of concurrent jobs over total number of jobs that were run during the chosen report period

- On the "load by time of day" chart this percent value is shown as a function of time of day (chart X axis scale: 24 hours; chart Y axis scale: 0 – 100%).
- On the "load by day of week" chart this percent value is shown as a function of day of week time of day (chart X axis scale: 168 hours which is 7 days * 24 hours in each day; chart Y axis scale: 0 – 100%).

Database Load Balancing Report (CPU Load)

This report is based on statistical analysis of job performance data collected by the Job Monitor. This report can be helpful if you have numerous database jobs running on your server. The Database Load Balancing Report can help you identify time windows with low server utilization and schedule new or modify existing jobs to run during these times.

The report contains two types of load charts: the "load by day of week" chart and the "load by time of day" chart. Both charts display database utilization as a percent of all jobs executed during the chosen report period.

Each chart is calculated using job performance data for the chosen report period. You can try different report periods. **Be aware that selecting longer report periods allows more job runs to be analyzed and thus it improves the report accuracy, but it can also increase the report calculation time.**

The following values are displayed on the report

Time – Time of Week (7 days, 24 hours in each day)

CPU Load – The CPU usage by all running jobs at a given point of time. The CPU Load is calculated as a percent of machine CPU time allocated for processing of active jobs over duration of these job runs.

- On the "load by time of day" chart the CPU Load value is shown as a function of time of day (chart X axis scale: 24 hours; chart Y axis scale: 0 – 100%, where 0 value means no job processing occurs and 100% means maximum job processing, in other words all CPU time is completely dedicated to the job processing and nothing is left for other processes).
- On the "load by day of week" chart the CPU Load value is shown as a function of day of week time of


day (chart X axis scale: 178 hours which is 7 days * 24 hours in each day; chart Y axis scale: 0 – 100%, where 0 value means no job processing occurs and 100% means maximum job processing, in other words all CPU time is completely dedicated to the job processing and nothing is left for other processes).

Important Notes:

- If the [Record System Statistics](#) option is not enabled, the Database Load Balancing Report does not take into account system statistics that could be collected by the Job Monitor. Both charts display only one CPU Load series for jobs statistics only, not including any other processing. You can compare this report against [Database Load Balancing Report](#) for applications, which is a part of the [DB Application Expert](#) to get the complete picture of your server load.
- If the [Record System Statistics](#) option is enabled, Both charts display 2 series – one series for the CPU load caused by jobs only and another series is for the overall database load including jobs and all other processing.

Top 10 Jobs by CPU Usage

This report presents several CPU usage charts. This report is based on statistical analysis of job performance data collected by the Job Monitor.

 **Important Note:** This reports requires that TIMED_STATISTICS option was enabled in the database during the selected report period

You can use this report for the following purposes:

- 1 To identify top 10 CPU consuming jobs. This information can help you in job troubleshooting.
- 2 This report also complements [Job Resource Consumption](#) report and both reports can be used to identify jobs that cause high system load. However, keep in mind that the Job Timing and Conflicts report charts job executing times, and NOT job CPU usage time. Long running jobs do not always take a lot of system resources.

The **Total CPU Usage** chart displays CPU usage (CPU time allocated) during the selected report period by all top 10 jobs.

The **Average CPU Load** chart displays average CPU usage during the selected report period for all top 10 jobs. The load is calculated as a ratio of job CPU time over job duration.

The **Average CPU Usage per Run** chart displays average CPU usage (CPU time allocated) during the selected report period for all top 10 jobs.

The **Maximum CPU Usage** chart displays high water mark for CPU usage (CPU time allocated) during the selected report period for all top 10 jobs.

The following values are displayed on the report

CPU Usage – seconds

CPU Load – as percent of job CPU time over job duration

Job ID – Unique job ID in the database

Legend – The chart legend describes job ID and job SQL.

Top 10 Jobs by Memory Usage

This report presents memory usage chart for jobs that ran during the selected report period. This report is based on statistical analysis of job performance data collected by the Job Monitor.

You can use this report for the following purposes:

- 1 To identify top 10 memory consuming jobs. This information can help you in job troubleshooting.
- 2 This report also complements [Job Resource Consumption](#) report and other Top 10 Jobs reports.

The following values are displayed on the report

Memory Load – The peak amount of memory allocated for the job session during job run. This includes both PGA and UGA memory.

Job ID – Unique job ID in the database

Legend – The chart legend describes job ID and job SQL.

Top 10 Jobs by Sorts

This report calculates 2 charts for jobs that ran during the selected report period. This report is based on statistical analysis of job performance data collected by the Job Monitor.

You can use this report for the following purposes:

- 1 To identify top 10 resource consuming jobs. This information can help you in job troubleshooting.
- 2 This report also complements [Job Resource Consumption](#) report and other Top 10 Jobs reports.

The **Top 10 Jobs by Number of Sorted Rows** chart displays number of sorted rows by job during the selected report period for top 10 jobs.

The **Sorts Detail** chart displays for every top job breakdown by sorts in memory vs. sorts on disk for top 10 jobs.

The following values are displayed on the report

Rows Sorted – The number of sorted rows.

Sorts (disk/memory) – The number of sort operations.

Job ID – Unique job ID in the database

Legend – The chart legend describes job ID and job SQL.

Top 10 Jobs by Disk I/O (Input/Output)

This report calculates 2 charts for jobs that ran during the selected report period. This report is based on statistical analysis of job performance data collected by the Job Monitor.

You can use this report for the following purposes:

- 3 To identify top 10 resource consuming jobs. This information can help you in job troubleshooting.
- 4 This report also complements [Job Resource Consumption](#) report and other Top 10 Jobs reports.

The **Top 10 Jobs by Disk I/O** chart displays top 10 jobs that caused most input output operations during the selected report period.

The **Job I/O Detail** chart displays for every top job breakdown by Logical I/O vs. Physical I/O that required disk access.

The following values are displayed on the report

I/O – The number of logical and physical input/output operations in thousands

Job ID – Unique job ID in the database

Legend – The chart legend describes job ID and job SQL.

DB Application Expert

Overview

DB Application Expert is an advanced database application monitoring and profiling tool. DB Application Expert is best used for monitoring front-end applications connected to Oracle databases. This tool compliments DB Job Expert that can be used to monitor Oracle back-end applications.

The graphical interface of DB Application Expert provides an enterprise view for quick, clear inspections of

- Applications working with the database
- Users who run these applications
- Computers that connect to the database server
- SQL executed by monitored applications
- Locks created by these applications and their impact on other applications
- Objects accessible by these applications
- Database performance and load analysis from an application point of view
- Many other application-level performance statistics that can be used for application analysis and troubleshooting

The back-end Application Monitor - monitoring and profiling package instantly monitors running applications and collects their performance statistics, resource usage, object access, SQL, locks and so on. The collected statistics are used in a variety of DB Application Expert reports that range from Application Locks and Conflicts to Top 10 Applications by CPU Usage and so on.

In addition to the Application Monitor package the DB Application Expert provides a set of back-end jobs that must be installed in the database in order to use advanced DB Application Expert features such as Application Profiler and application performance reports. The Application Monitor package is used to monitor active applications and collect application performance statistics. The Application Profiler uses collected application performance data in most DB Application Expert Reports. The collected data can be also used as a complete application audit log.

DB Application Expert is built around Oracle scheduling API provided in the standard DBMS_JOB package.

Before you start using DB Application Expert back-end components and jobs, make sure you have properly configured your Oracle job queues and SNP background processes that actually run jobs. You must have at least one SNP process running to execute your scheduled and queued jobs. Note that SNP processes will not execute job queues if the database is in restricted mode. An instance can have up to 36 SNP background processes.

In order to use job queues you will need to set the following two INIT.ORA parameters:

1. JOB_QUEUE_PROCESSES = n

Here n is how many SNP background processes you want running.

Default: 10

Valid range of values: 0...36

2. JOB_QUEUE_INTERVAL = n

Where n is the interval between wake-ups, in seconds, for the SNP background processes.

Default: 60

Valid range of values: 0...3600

Please be aware that if either JOB_QUEUE_PROCESSES = 0 or JOB_QUEUE_INTERVAL = 0, scheduled Application Monitor jobs will not run

See the following topics for more information on DB Application Expert

[Application Monitor Console](#)

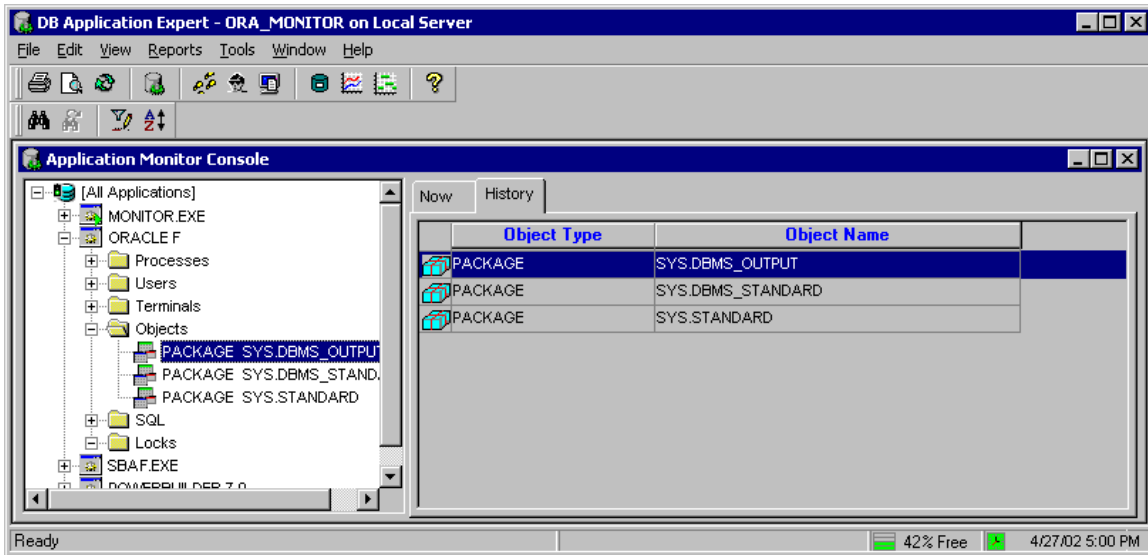
[Application Monitor Installation](#)

[Application performance statistics and Application Monitor Options](#)
[Application Performance Reports](#)

Application Monitor Console

The Application Monitor Console is a visual monitoring tool you can use to inspect currently active applications and historical statistics collected by the back-end Application Monitor package.

The Application Monitor Console is automatically started when DB Application Expert GUI is started.



Navigation Techniques

The Application Monitor Console allows two different ways of viewing and application statistics:

The tree-view on the left-hand of the console window presents applications grouped by the entry-point application executable file name.

The grid-view displays various applications statistics. The right-pane has two tab pages: **Now** – statistics for currently active applications and **History** – historical statistics for all applications that have run in the database since the Application Monitor back-end job was installed and activated or since the last time Application Monitor repository tables were reset whatever was last. To view various groups of statistics select the appropriate statistics group on the left pane.



Tips:

Select **[All Applications]** item on the left pane and then switch to the **History** tab on the right pane to view averages and totals of various application performance parameters for all database applications. This view can help you see which application(s) are most common and how they affect your database performance.

Select the required application name on the left pane and then switch to the **History** tab on the right pane to view detailed statistics for each run of the selected application

Select the required group of statistics (such as Locks, Users, etc...) for the required application on the left pane and then switch to the **History** tab on the right pane to view detailed information for the selected category.

You may access navigation controls through the main menus, at the top of the Application Monitor Console window; through the toolbar, which allows you to manipulate applications in the tree-view; or through the context-sensitive menu that appears when you right-click an item in the tree-view or in the grid-view. All

methods allow you to choose such functions as print, zoom, expand, refresh, filter or sort, and other. The set of enabled functions depends on the type of selected item and may differ for different items.

Remember that while the main menu contains almost all controls and functions available in DB Application Expert, only the context-sensitive menus allow performing context operations such as Print and Print Preview. In other words, if you right-click the tree-view and then select the **Print** command from the popup context menu, this will print the contents of the tree-view, but if you right-click the grid-view and then select the **Print** command from the popup context menu, this will print the contents of the grid view.

Application Monitor Installation

To install Application Monitor use the graphical DB Tools [Database Setup utility](#). Follow instructions provided by the Setup Wizard.

Application Monitor is built around Oracle scheduling API provided in the standard DBMS_JOB package.

Before you start using DB Application Expert back-end components and jobs, make sure you have properly configured your Oracle job queues and SNP background processes that actually run jobs. You must have at least one SNP process running to execute your scheduled and queued jobs. Note that SNP processes will not execute job queues if the database is in restricted mode. An instance can have up to 36 SNP background processes.

In order to use job queues you will need to set the following two INIT.ORA parameters:

1. JOB_QUEUE_PROCESSES = n

Here n is how many SNP background processes you want running.

Default: 10

Valid range of values: 0...36

2. JOB_QUEUE_INTERVAL = n

Here n is the interval between wake-ups, in seconds, for the SNP background processes.

Default: 60

Valid range of values: 0...3600

Please be aware that if either JOB_QUEUE_PROCESSES = 0 or JOB_QUEUE_INTERVAL = 0, scheduled Application Monitor jobs will not run

See also: [Application performance statistics and Application Monitor Options](#) topic.

Application Performance Statistics and Application Monitor Options

Application Performance Statistics

Application Monitor collects statistics for every running application that has duration less or equal to restart interval of the **app_monitor.start_monitor** job scheduled to run in the database. By default the job restart interval is 10 seconds, which should suite in most environment, does not cause any significant load on the database and be short enough for monitoring any significant application. You can change this job restart interval using the [DB Job Expert](#). High values may result in poor statistics accuracy but cause insignificant CPU consumption, on the other hand low values may result in good statistics accuracy but cause significant CPU consumption. There is no recommended value for the restart value because the optimal value of this

parameter is highly depended on your database system usage, hardware and software platform and many other factors. Use the DB Job Monitor to determine whether the application monitor job creates too much load on the database then you can anticipate. Try to keep this value as small as possible without causing much load.

The Application Monitor can also optionally collect overall system statistics during application runs and during periods of time when no application is running.

You can choose and pick which application performance statistics you want to collect using the Application Monitor Options (Tools/Application Monitor Options menu). The collected statistics are used in a variety of DB Application Expert reports. Collected system statistics can be used in custom build reports (you can use the Performance Library tools) to find windows in server heavy processing so you can better balance database load created by these application, back-end database jobs and other processes.

All collected statistics are written into the following Application Monitor repository tables:

```
ORA_MONITOR.APP_STATS
ORA_MONITOR.APP_NAMES
ORA_MONITOR.APP_LOCKS
ORA_MONITOR.APP_LOCK_WAITS
ORA_MONITOR.APP_OBJECTS
ORA_MONITOR.APP_SQL
ORA_MONITOR.APP_STATS_CONFIG
```

Not all statistics are available by default. Statistics such as **CPU usage** are available only when Oracle **TIMED_STATISTICS** option is turned on. If you want to collect CPU and other timed statistics you must turn on **TIMED_STATISTICS** option in INIT.ORA or each time the system starts turn them on using the following command

```
ALTER SYSTEM SET timed_statistics=TRUE;
```

Application Monitor Options

To change Application Monitor parameters use **Tools/Application Monitor Options** menu in the DB Application Expert GUI

The following parameters can be changed.

Record Performance Statistics (Yes/No) -- This parameter controls whether you want to collect general application performance statistics such as CPU and memory usage, run-times, number of performed sorts, number of created logical and physical I/O operations and other similar statistics.

Record Users (Yes/No) -- This parameter controls whether you want to collect name of database and operating system users who run database applications and also names of computers where these front-end application are installed and run.

Record Objects (Yes/No) -- This parameter controls whether you want to collect names and types of database objects accessible by applications

Record Locks (Yes/No) -- This parameter controls whether you want to monitor and collect locks created by database applications including how these locks affect other applications and processes.

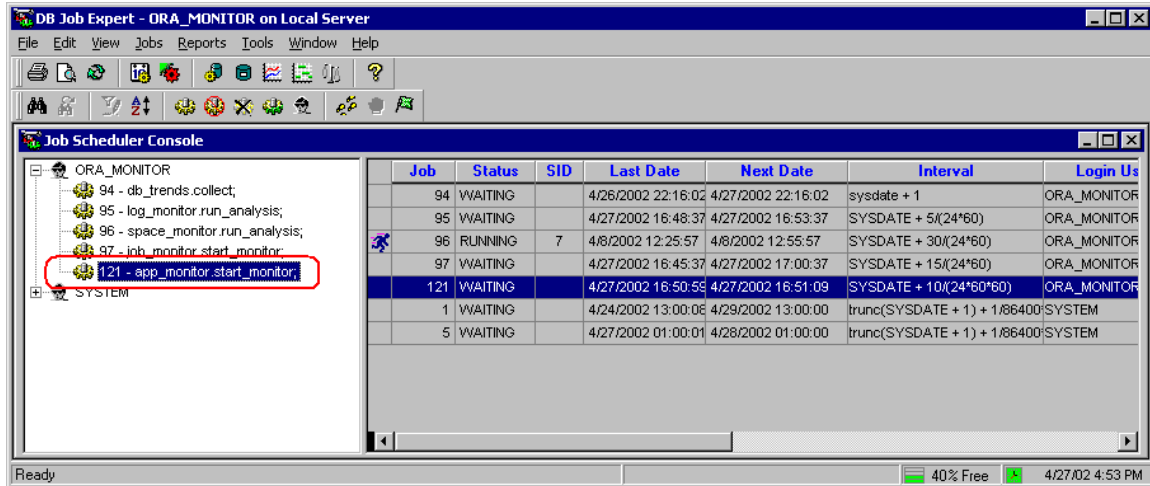
Record Users (Yes/No) -- This parameter controls whether you want to collect name of database and operating system users who run database applications and also names of computers where these front-end application are installed and run.

Record SQL (Yes/No) -- This parameter controls whether you want to collect application SQL statements.

Record System Statistics (Yes/No) -- This parameter controls whether the Application Monitor should also record system statistics between and during application runs. This information can be used in finding windows for application runs and other application performance analyses.



Important Note: The statistics collected by the Application Monitor are as accurate as accurate are settings for the Application Monitor job (see **app_monitor.start_monitor** job on the following picture). If the job settings are incorrect it may lead to a situation when the Application Monitor will miss important applications, application statistics and/or SQL statements executed by these applications.



Application Performance Reports

DB Application Expert currently supports 11 different application performance and application troubleshooting reports. Each report can be viewed and printed from the Report window. These reports are based on the application performance statistics collected by the [Application Monitor package](#).

The following reports are available:

- [Application Profiler Statistics \(by Application\)](#)
- [Application Profiler Statistics \(by User\)](#)
- [Application Profiler Statistics \(by Terminal / Workstation\)](#)
- [Application Resource Consumption](#)
- [Application Performance & Trends](#)
- [Application Locks and Conflicts](#)
- [Database Load Balancing](#)
- [Top 10 Applications by CPU Usage](#)
- [Top 10 Applications by Memory Usage](#)
- [Top 10 Applications by Sorts](#)
- [Top 10 Applications by Disk I/O](#)

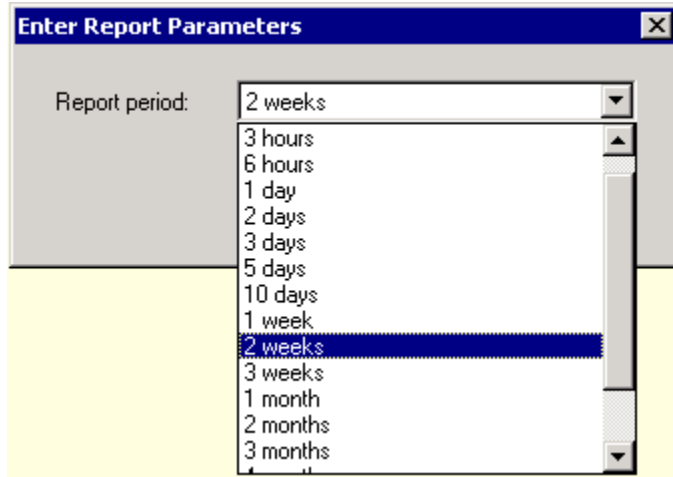
You can use them to monitor applications and users, spot application trends, tune and troubleshoot applications, troubleshoot database locks and resource usage, and provide others with information through printing. Click on a report name in the list above for more information.

Tips:

- Most reports use application performance data for application processing analysis. This data is not available automatically. To allow application performance data collection you must install the back-end [Application Monitor](#) package and job. Do not run reports immediately after you install these back-end components. Wait as long as necessary for each application to run at least 4-5 times. This will allow the Application Monitor to gather sufficient volume of application performance data for an accurate application analysis. Do not force/trigger applications manually for the purpose of performance data collection, as it will lead to inaccurate reports. The accuracy of reports is proportional to the volume of gathered data. Longer you wait, more applications run, and more valuable application performance data can be gathered.
- Application performance data is stored in the ORA_MONITOR.APP_STAT table. If for some reasons you cannot find a pre-built application report that fits your needs, feel free to create your own reports using the [Performance Library](#) tools. For description and usage of Application Monitor repository tables see source code for the Application Monitor package provided with DB Tools.


Viewing reports:

- 4 From the Report menu, choose the type of report you want. The report window will appear.
- 5 If the selected report analyses historical application performance data, you will be prompted to choose time interval for the analysis. Select the required interval from the drop-down list and click OK.




- 6 You can use the Zoom feature to zoom in and out. You can either select one of the predefined values in the zoom drop-down box or type in the value you want then press the Enter key.

Printing reports:

- 3 Run report as described above.
- 4 When the report is ready, click the Print  toolbar button or select **File/Print** command from the main menu

Refreshing reports data:

- 1 Click the Refresh  toolbar button or select **View/Refresh** command from the main menu.
- 2 If the selected report analyses application performance data, you will be prompted to choose time interval for the analysis. Select the required interval from the drop-down list and click OK.

Application Resource Consumption Report

This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

This report shows trend chart for the total system load created by all applications that ran during the selected report period and also shows the system load created by individual applications. You can use this report for the following purposes:

- 3 To identify applications consuming lots of system resources.
- 4 To evenly distribute your system load between different applications so that the resource intensive applications do not run at the same time. This information can help you in improving the overall system performance, improving application response times and meeting required service levels.

The chart is scaled to fit the selected report period. You can try different report periods to see how the system is loaded over different periods of time. **Be aware that selecting longer report periods allows more application runs to be analyzed and thus it improves the report accuracy, but it can also increase the report calculation time.**

The following columns and values are displayed on the report

CPU Usage (chart) – Total CPU time spent for executing applications

Time (chart) – System time

Application ID – The unique pseudo ID of a monitored application

Program Name – The name of the main application program module (as reported by Oracle)

Total Executions – Number of times the application was executed during the selected report period

Average CPU Time – The average CPU time spent per application run during the selected report period

Total CPU Time - Total CPU time spent for all application runs during the selected report period



Tips:

It is a good idea to run [Application Locks and Conflicts](#), which together with Job Resource Consumption Report will provide you with a complete picture of the system load and thus allow you to efficiently allocate system's CPU time and other resources.



Note:

Applications whose execution took less than 1 second of CPU time are NOT included in this report

Application Performance and Trends Reports

This report presents trend-line charts for top 20 applications whose average run-time duration have changed during the selected report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report for the following purposes:

- 3 Identifying applications whose run-time duration significantly changed from run-to-run during the chosen report period.
- 4 Identifying application duration trends (i.e., how quickly the duration changes over time) and thus you can use this information to forecast future durations.

You should choose a meaningful time interval for the report to allow the analyzed applications to run for at least 4-5 times. Keep in mind that the trend-chart accuracy for each application highly depends on the number of successful application runs during the chosen report period. Applications with less than 3 executions during the chosen report period are ignored when the report is built. The chart is scaled to fit the selected report period. You can try different report periods. **Be aware that selecting longer report periods allows more application runs to be analyzed and thus it improves the report accuracy, but it can also increase the report calculation time.**

The following values are displayed on the report

Duration – Application duration in seconds

Time – System time

Legend – Chart legend that shows up to 20 applications presented on the chart. Each legend label contains name of the main application executable module (as reported by Oracle).

Colored solid lines on the chart indicate actual application durations over time. Colored dashed lines on the chart indicate application duration trends. Application trend lines are built using simple linear regression model. Both actual and trend lines are drawn using the same unique color assigned to each application.

Application Locks and Conflicts Report

This report presents all locks that cause other processes to wait. The report includes only locks recorded during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report for the following purposes:

- 1 To identify concurrent or conflicting applications, those that happen to run at the same time or delay other processes. This information can help you in application troubleshooting.
- 2 This report also complements [Application Resource Consumption](#) report and both reports can be used to identify applications that cause high system load and affect other processing. However, keep in mind that the Application Locks and Conflicts report shows lock times and durations. Long running applications and long locks do not always take a lot of system resources and/or affect other applications.

The following values are displayed on the report

Locking Program – The name of the main application executable program (as reported by Oracle) that creates locks holding other processes.

Program – The name of the main application executable program (as reported by Oracle) that was waiting to gain access to object locked by the **Locking Program**.

Start Time – The start time when the **Program** begin waiting to acquire access to the locked object(s).

End Time – The start time when the lock was released and the **Program** stopped waiting to acquire access to the locked object(s).

Duration – The duration of the wait.

Object Type – The type of the object being locked by the **Locking Program**.

Object Name – The full name of the object being locked by the **Locking Program**.

Database Load Balancing Report

The Database Load Balancing Report can help you identify time windows of high and low server utilization and setup new or modify existing applications to run during low utilization times. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

The report contains two system load charts: the "load by day of week" chart and the "load by time of day" chart. Both charts display summary database utilization (% of CPU time) for all applications executed during the chosen report period.

Each chart is calculated using application performance data for the chosen report period. You can try different report periods. **Be aware that selecting longer report periods allows more application runs to be analyzed and thus it improves the report accuracy, but it can also increase the report calculation time.**

The following values are displayed on the report

Time – Time of day (1 day scale, 24 hours)

CPU Load – The CPU usage by all running applications at a given point of time. The CPU Load is calculated as a percent of machine CPU time allocated for processing of active applications over duration of these application runs.

- On the "load by time of day" chart the CPU Load value is shown as a function of time of day (chart X axis scale: 24 hours; chart Y axis scale: 0 – 100%, where 0 value means no application processing performed by the database and 100% means the maximum theoretical value for the CPU usage, in other words all CPU time is completely dedicated to the application processing and nothing is left for other processes).
- On the "load by day of week" chart the CPU Load value is shown as a function of day of week time of day (chart X axis scale: 178 hours which is 7 days * 24 hours in each day; chart Y axis scale: 0 – 100%, where 0 value means no application processing occurs and 100% means maximum application processing, in other words all CPU time is completely dedicated to the application processing and nothing is left for other processes).

**Important Note:**

- If the [Record System Statistics](#) option is not enabled, the Database Load Balancing Report does not take into account system statistics that could be collected by the Application Monitor. Both charts display only one CPU Load series for application statistics only, not including any other processing. Compare this report against [Database Load Balancing Report \(CPU Load\)](#) for jobs, which is a part of the [DB Job Expert](#) to get the complete picture of your server load.
- If the [Record System Statistics](#) option is enabled, Both charts display 2 series – one series for the CPU load caused by applications only and another series is for the overall database load including applications and all other processing.

Top 10 Applications by CPU Usage

This report provides several different views focusing on top 10 applications that have consumed most CPU during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).



Important Note: This reports requires that TIMED_STATISTICS option was enabled in the database during the analyzed report period

You can use this report for the following purposes:

- 1 To identify top 10 CPU consuming applications. This information can help you in application troubleshooting.
- 2 Together with the [Application Resource Consumption](#) report it can be used to identify applications causing high system loads. However, keep in mind that the Application Resource Consumption report charts application executing times, and NOT application CPU usage time. Long running applications do not always take a lot of system resources.

The **Total CPU Usage** chart displays CPU usage (CPU time allocated) during the selected report period by all top 10 applications.

The **Average CPU Load** chart displays average CPU usage during the selected report period for all top 10 applications. The load is calculated as a ratio of application CPU time over application duration.

The **Average CPU Usage per Run** chart displays average CPU usage (CPU time allocated) during the selected report period for all top 10 applications.

The **Maximum CPU Usage** chart displays high water mark for CPU usage (CPU time allocated) during the selected report period for all top 10 applications.

The following values are displayed on the report

CPU Usage – seconds

CPU Load – as percent of application CPU time over application duration

Program – The first 10 characters of name of the main application executable program (as reported by Oracle).

Legend – The chart legend describes full program names.

Top 10 Applications by Memory Usage

This report calculates memory usage chart for applications that ran during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report for the following purposes:

- 1 To identify top 10 memory consuming applications. This information can help you in application troubleshooting.
- 2 This report also complements [Application Resource Consumption](#) report and other Top 10 Applications

reports.

The following values are displayed on the report

Memory Load – The peak amount of memory allocated for the application session during application run. This includes both PGA and UGA memory.

Program – The first 10 characters of name of the main application executable program (as reported by Oracle).

Legend – The name of the main application executable program (as reported by Oracle)

Top 10 Applications by Sorts

This report calculates 2 charts for applications that have run during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report for the following purposes:

- 1 To identify top 10 resource consuming applications. This information can help you in application troubleshooting.
- 2 This report also complements [Application Resource Consumption](#) report and other Top 10 Applications reports.

The **Top 10 Applications by Number of Sorted Rows** chart displays number of sorted rows by application during the selected report period for top 10 applications.

The **Sorts Detail** chart displays for every top application breakdown by sorts in memory vs. sorts on disk for top 10 applications.

The following values are displayed on the report

Rows Sorted – The number of sorted rows.

Sorts (disk/memory) – The number of sort operations.

Program – The first 10 characters of name of the main application executable program (as reported by Oracle).

Legend – The name of the main application executable program (as reported by Oracle)

Top 10 Applications by Disk I/O (Input/Output)

This report calculates 2 charts for applications that have run during the chosen report period. This report is based on statistical analysis of application performance data collected by the [Application Monitor](#).

You can use this report for the following purposes:

- 1 To identify top 10 resource consuming applications. This information can help you in application troubleshooting.
- 2 This report also complements [Job Resource Consumption](#) report and other Top 10 Applications reports.

The **Top 10 Applications by Disk I/O** chart displays top 10 applications that caused most input output operations during the selected report period.

The **Job I/O Detail** chart displays for every top application breakdown by Logical I/O vs. Physical I/O that required disk access.

The following values are displayed on the report

I/O – The number of logical and physical input/output operations in thousands

Program – The first 10 characters of name of the main application executable program (as reported by Oracle).

Legend – The name of the main application executable program (as reported by Oracle)

Application Profiler Statistics (by Application)

This report shows application run statistics collected by the Application Monitor and Profiler. The statistics are grouped by application program names. Each group includes program start time, run duration, name of the database user, name of the operation system user running the application, terminal and/or workstation name and other important performance parameters obtained from the V\$SESSION dynamic performance view for each reported application run. For more information on the performance parameters please refer to your Oracle Database Administration Guide.

See also:

[Application Profiler Statistics \(by User\)](#)

[Application Profiler Statistics \(by Terminal / Workstation\)](#)

Application Profiler Statistics (by User)

This report shows application run statistics collected by the Application Monitor and Profiler. The statistics are grouped by database user names. Each group includes program start time, run duration, name of the database user, name of the operation system user running the application, terminal and/or workstation name and other important performance parameters obtained from the V\$SESSION dynamic performance view for each reported application run. For more information on the performance parameters please refer to your Oracle Database Administration Guide.

See also:

[Application Profiler Statistics \(by Application\)](#)

[Application Profiler Statistics \(by Terminal / Workstation\)](#)

Application Profiler Statistics (by Terminal / Workstation)

This report shows application run statistics collected by the Application Monitor and Profiler. The statistics are grouped by names of terminals and workstations on which users run their database applications. Each group includes program start time, run duration, name of the database user, name of the operation system user running the application, terminal and/or workstation name and other important performance parameters obtained from the V\$SESSION dynamic performance view for each reported application run. For more information on the performance parameters please refer to your Oracle Database Administration Guide.

See also:

[Application Profiler Statistics \(by Application\)](#)

[Application Profiler Statistics \(by User\)](#)

Test Data Generator

Test Data Generator Overview

Test Data Generator is a perfect tool for creating large volumes of meaningful data for performance testing, quality assurance testing, usability analysis, and many other purposes.

Test Data Generator is able to automatically detect primary and foreign keys and without user intervention generate test data that match all referential integrity constraints rules.

Test Data Generator can generate test data for all standard Oracle data types. You can customize the default data generation rules on per-table-column basis to fit your specific business requirements.

Test Data Generator comes with a large set of pre-built data libraries of the most commonly used data values including meaningful addresses, names, phone numbers, company names, industries, financial terms, healthcare data, drugs, auto-parts and more. It also gives you the flexibility to create and modify your own data libraries, create custom data generation rules, and specify existing database tables with real business information as "lookup" data values for the newly generated data.

See also:

- [Working with Test Data Generator](#)
- [Choosing Data Generation Method](#)

Working With Data Generator

Test Data Generator uses the concept of a "run" for populating a single database table with a test data. A run involves running the data generation process and inserting data to the selected table. To simplify data generation process for multi-table environments Test Data Generator provides support for projects. Data Generator projects allow grouping of related tables and persistently storing project configuration and data generation rules in the database along with the development comments.

Opening an Existing Project for Modifications

1. Select **File/Open Project** menu (CTRL O keyboard shortcut). The **Open Project** dialog will appear. Only projects associated with your user id will be displayed in the project tree.
2. Select the project you want to open and then click the **Open** button. The **Project Tables** dialog will appear. Continue as needed.

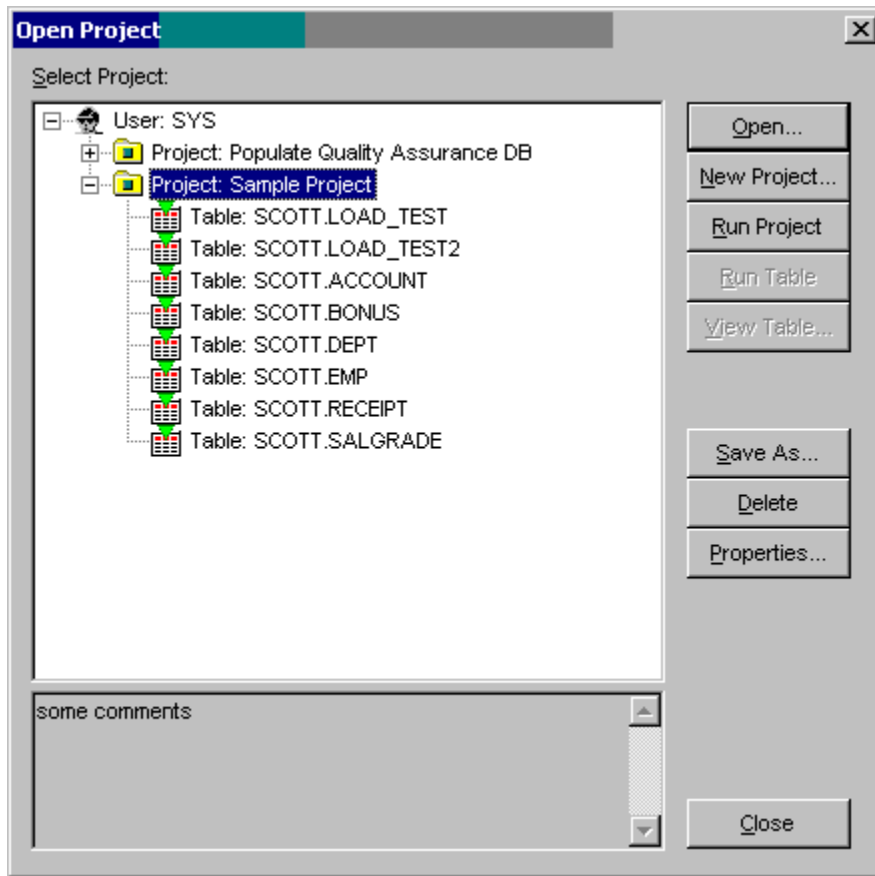
Creating a New Project

In case if the **Open Project** dialog is not displayed:

1. Select **File/New Project** menu (CTRL N keyboard shortcut). The **New Project** dialog will appear.
2. Enter project name and optional comments. Although it is not required but it is highly recommended to use unique project name (project name should be unique for your user id). Click the **OK** button. The **Project Tables** dialog will appear. Continue as needed.

In case if the **Open Project** dialog is displayed:

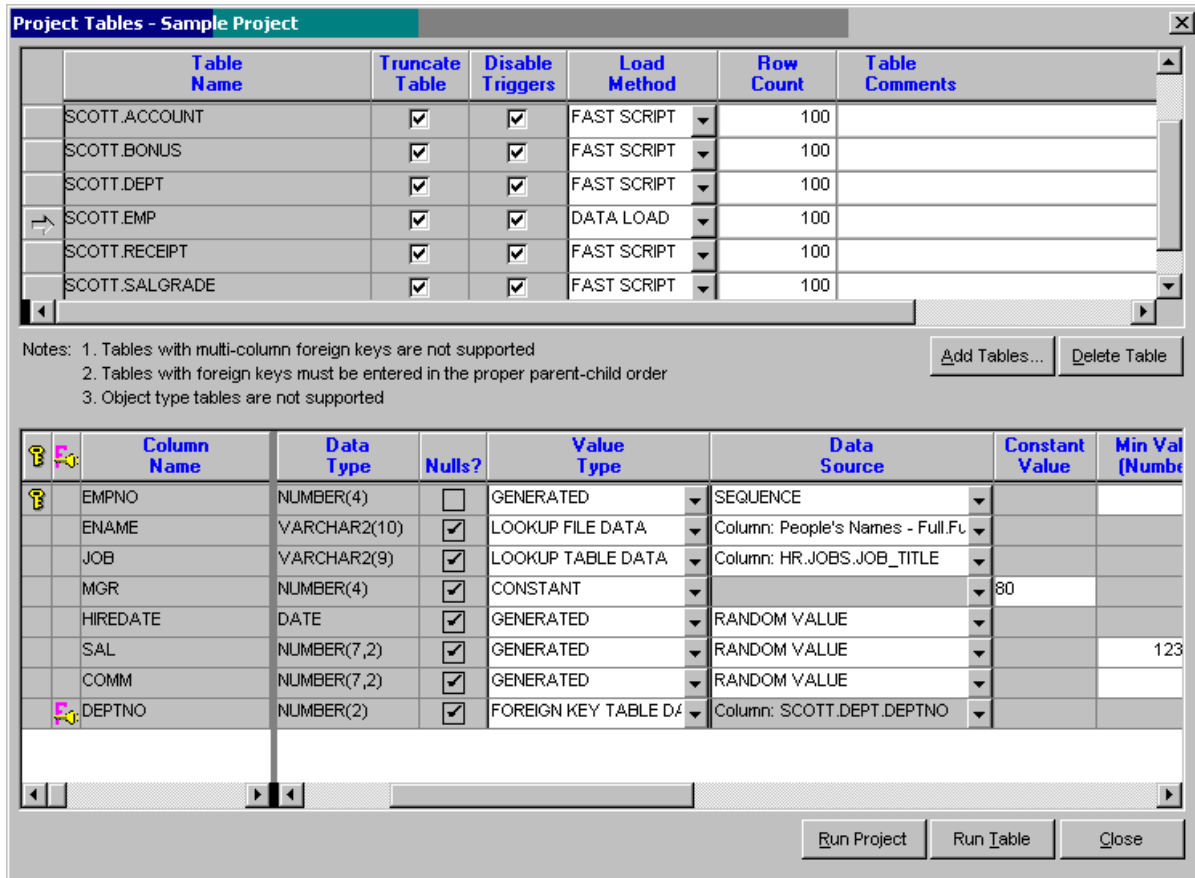
1. Click the **New Project** button The **New Project** dialog will appear.
2. Enter project name and optional comments. Although it is not required but it is highly recommended to use unique project name (project name should be unique for your user id). Click the **OK** button. The **Project Tables** dialog will appear. Continue as needed.



Open Project dialog window

New project settings and all consecutive changes will be saved in the Test Data Generator's repository tables. If these tables do not exist at the time you create your first project Test Data Generator will prompt your permissions to install them. The repository tables can be also installed at any time using DB Tools [Database Setup utility](#).

The following image contains screenshot of a sample Test Data Generator project



Deleting an Existing Project

1. Select **File/Open Project** menu (CTRL O keyboard shortcut) to open an existing project. The **Open Project** dialog will appear.
2. Select the project you want to delete and then click the **Delete** button. The Please Confirm message box will appear. Click the YES button to confirm the deletion. This will delete the project and its settings from the Test Data Generator repository tables.

Adding Tables to a Project

1. Select **File/Open Project** menu (CTRL O keyboard shortcut) to open an existing project. The **Open Project** dialog will appear.
2. Select the project you want to modify and then click the **Open** button. The **Project Tables** dialog will appear.
3. Click the Add Tables button to add one or more tables to the opened project. The [Select Tables](#) dialog will appear. Select one or more tables in that dialog and then click the OK button. The dialog will close and all selected tables will appear in the table list.

Removing Tables from a Project

1. Select **File/Open Project** menu (CTRL O keyboard shortcut) to open an existing project. The **Open Project** dialog will appear.

2. Select the project you want to modify and then click the **Open** button. The **Project Tables** dialog will appear.
3. Click on the name of the table you want to remove from your project.
4. Click the Delete Table button.
5. If you want to remove



Important Note: Removing a table from Test Data Generator's project does not physically destroy table or table data in the database. It simply deletes appropriate records from the Test Data Generators' repository tables.

Running Data Generation and Loading Process

1. Select **File/Open Project** menu (CTRL O keyboard shortcut) to open an existing project. The **Open Project** dialog will appear.
2. Select the project you want to run
3. Click the **Run Project** button.

In case if you run or rerun just one table select

1. Select **File/Open Project** menu (CTRL O keyboard shortcut) to open an existing project. The **Open Project** dialog will appear.
2. Select the project whose table you want to run and expand the project tree
3. Select the required table
4. Click the **Run Table** button.

Table and Project data generation process can be also started from the [Project Tables](#) screen.

Data Generation Methods and Options

Truncate Table

This table-level option instructs Test Data Generator to truncate table data before inserting new test data.

Disable Triggers

This table-level option instructs Test Data Generator to temporarily disable all table triggers before inserting new test data. All triggers are enabled back after the data generation process is complete.

Disabling triggers can improve data generation process the performance as well as avoid applying unnecessary business logic to the newly inserted fresh data if such logic is provided in the database triggers



Important Note: Disabling table triggers only affects the table being populated with the test data. It does not affect other tables in the database and it does not disable Oracle triggers facility.

Number of rows

This option controls how many rows are inserted into the table during test data generation process.

Load method

This option controls how test data is generated and inserted in to the target table. Two methods are presently supported:

- **FAST SCRIPT** – this method instructs Test Data Generator to create dynamic stored procedure for every selected table whose purpose is to generate the test data according to selected rules and insert it into the table.
- **DATA LOAD** – this method instructs Test Data Generator to generate and write test data into a disk file then load the resulting file into the table.



Note: Because the loading method is specified on the table level you can pick different methods for different tables.

Considerations for choosing optimal loading method

FAST SCRIPT Pros:

1. FAST SCRIPT is the most efficient for generating large volumes of data. This method is lightning fast; usually 100,000 records can be created and inserted in a matter of several seconds.

FAST SCRIPT Cons:

1. FAST SCRIPT requires Oracle's DBMS_RANDOM package that must be installed in the database. The mentioned package is normally available in Oracle database versions 8i and up.
2. Data generation process using FAST SCRIPT method cannot be interrupted using the front-end interface. The only way to stop the process is to find and kill the database session.
3. FAST SCRIPT method runs completely on the back-end and because of that it cannot access Test Data Generator's [data libraries](#) stored in data files on the front-end side.

DATA LOAD Pros:

1. DATA LOAD supports any Oracle database versions starting with 7.0 and up
2. Data generation process can be interrupted at any time by clicking the Cancel button on the Progress dialog.
3. Rich [data libraries](#) can be used to generate realistic test data.

DATA LOAD Cons:


1. DATA LOAD is not efficient as FAST SCRIPT and requires significantly more time to populate large volume tables because the data is first written to temporary files and then these files are inserted record-by-record into their target tables.
2. DATA LOAD can lead to significant network traffic when sending millions of test records over the SQL*Net connection to the database.

Column-level Options

Column-level data generation options and method vary for different column types. The following table describes supported options:

Option Name	Choices	Numeric Data Types	String Data Types	Date/Time Data Types	Comments
Value Type	GENERATED	√	√	√	Value is generated randomly or sequentially using random number /text generator
	CONSTANT	√	√	√	User-defined constant value inserted into all table rows
	NULL	√	√	√	Insert NULL
	FOREIGN KEY TABLE DATA	√	√	√	Lookup values in the referenced table. Lookup is based on the random selection by picking random row ID in the referenced table.
	LOOKUP TABLE DATA	√	√	√	Look values in the user-defined lookup table and column. Lookup is based on the random selection by picking random row ID in the specified table.

	LOOKUP FILE DATA	√	√	√	Lookup values in the user-picked Data Generator's data library
Data Source	**The choice depends on the selected Value Type	√	√	√	<ul style="list-style-type: none"> When Value Type is GENERATED you can pick either RANDOM or SEQUENCE as value data source. RANDOM guarantees that generated values are randomly generated using uniform distribution model. SEQUENCE as value type guarantees that generated values are produced as uninteruptible set of sequential numbers or dates or text values (text values are ended with a sequential numeric suffix). When Value Type is CONSTANT or NULL this option is not used. When Value Type is FOREIGN KEY TABLE DATA this option is fixed and cannot be changed. The value is always retrieved from the referenced table and columns whose names are displayed in <schema>.<table>.<column> format. When Value Type is LOOKUP TABLE DATA you can pick any existing table and column in the database as a data source. The selected lookup reference is displayed in <schema>.<table>.<column> format. When Value Type is LOOKUP FILE DATA you can pick a column in any appropriate Data Generator's data library. The selected lookup reference is displayed in <file>.<column> format.
Constant Value	User-defined	√	√	√	User-defined constant value
Min Value (Number)	User-defined	√			You can use Min/Max values to define range of generated values. This option is only valid when the Value Type is GENERATED.
Max Value (Number)	User-defined	√			You can use Min/Max values to define range of generated values. This option is only valid when the Value Type is GENERATED and the Data Source is RANDOM.
Min Value (Date)	User-defined			√	You can use Min/Max values to define range of generated values. This option is only valid when the Value Type is GENERATED.
Max Value (Date)	User-defined			√	You can use Min/Max values to define range of generated values. This option is only valid when the Value Type is GENERATED and the Data Source is RANDOM.
Include Time	Yes/No			√	Instructs Test Generator to add random time to generate date values. This option is only valid when the Value Type is GENERATED and the Data Source is RANDOM.
Prefix Text	User-defined		√		User-defined fixed leading text for all generated values. This option is only valid when the Value Type is GENERATED.
Suffix Text	User-defined		√		User-defined fixed trailing text for all generated values. This option is only valid when the Value Type is GENERATED.
Text Case	UPPER LOWER MIXED		√		Controls text case for generated values. This option is only valid when the Value Type is GENERATED.
Text Length	User-defined		√		Overrides maximum length of the data value (as compared to the column size). When generating test data the Data Generator ensures that produced text values are within the specified length limit. This option is only valid when the Value Type is GENERATED.

Format	User-defined		√		Controls formatting applied to the generated number before it is inserted into the database. If formatting mask includes at least one # symbol Data Generator applies the specified format mask to randomly or sequentially generated values. # symbols are used for numeric values, @ symbols are used for alpha values. For example, to generate telephone numbers in (nnn) nnn-nnnn format specify format mask as (###) ###-####. To generate product codes in 12ABC-XYZ format specify 12 for the Prefix Text option, UPPER for the Text Case option and finally specify @@@-@@@ for the Format mask option. This option is only valid when the Value Type is GENERATED.
Space Padding	LEFT NONE RIGHT		√		Controls space padding. If this option is not "NONE" and the length of the generated value is less then the number in Text Length column then DB Tools automatically appends or prefixes additional spaces to the generated value in order to make it long enough This option is only valid when the Value Type is GENERATED.
Percent of Nulls	0...100	√	√	√	Instructs Data Generator to produce certain percent of NULL values. By default Data Generator always produces not NULL values. This option is not valid is the column features NOT NULL check constraint.  Note: The amount of inserted NULL values is not guaranteed to exactly match user-defined Percent of Nulls

Data Libraries

Data Libraries are table-like files containing sets of commonly used values that you can use as data sources during the generation process.

Test Data Generator allows you to pick any data column from any library as a data source for the value in a column of the target table. In other words the set of columns in a library file does not need to match columns in the target table because you specify value source for each column separately. Because of that a library file containing 4 columns and 200 unique records can be used to produce 1552438800 data records from the. Throwing in other columns with random or sequential values in the target table allows Test Data Generator to produce virtually infinite number or records in any table. The real limitations here are really only data type size constraints and business rules enforced by relational table constrains. For example if a column in a single column table has data type NUMBER(3) and there is a unique constraint, index or primary key defined on the table then only 999 unique rows can be inserted into the table (if not taking into account rows with 0 and negative values).

The following data libraries are included in the software distribution package:

1. **Airport Data (total 267 unique records):**
 - Airport Name
 - Airport Code
 - Country Name
 - Country Code
2. **Auto parts (total 171 unique records)**
 - Part Name
3. **Books – Oracle related (total 157 unique records)**

- Author Name(s)
 - Book Name
 - Cover Type
 - ISBN Number
4. **Books by William Shakespeare (total 199 unique records)**
 - Author Name(s)
 - Book Name
 - Cover Type
 - ISBN Number
 5. **Fortune 500 Company Names (total 500 unique records)**
 - Company Name
 6. **Computer Science Magazines (total 154 unique records)**
 - Publication Name
 7. **Countries (total 240 unique records)**
 - 2-character Country Code
 - Country Name
 8. **Denmark Cities (total 82 unique records)**
 - City Name
 9. **Denmark States (total 16 unique records)**
 - State Name
 10. **European Colleges (total 1283 unique records)**
 - College Name
 11. **FDA Drugs and Components (total 3000 unique records)**
 - Application #
 - Product #
 - Form
 - Dosage
 - Product Marketing Status
 - TE Code
 - Reference Drug
 - Drug Name
 - Active Ingredient
 12. **French Postal Codes (total 44 unique records)**
 - City Name
 - Postal Code
 13. **Medical Procedures (total 248 unique records)**
 - Procedure Code
 - Procedure Description
 14. **Microprocessors (total 567 unique records)**
 - Processor Name
 15. **New York Streets (total 794 unique records)**
 - Street Name
 16. **People's Names - First and Last Name separately (total 500 unique records)**
 - First Name

- Last Name

17. People's Names – Full Name (total 500 unique records)

- Full Name

18. Periodic Table of Chemical Elements (total 109 unique records)

- Symbol
- Element Name

19. Sample Customers (total 99 unique records)

- Customer ID
- Company Name
- Contact Name
- Contact Title
- Address
- City
- Region
- Postal Code
- Country
- Phone
- Fax

20. Sample Orders (total 878 unique records)

- Order ID
- Customer
- Order Date
- Required Date
- Shipped Date
- Ship Via
- Freight
- Ship Name
- Ship Address
- Ship City
- Ship Region
- Ship Postal Code
- Ship Country

21. Sample Products (total 109 unique records)

- Product ID
- Product Name
- Supplier
- Category
- Quantity Per Unit
- Unit Price
- Units In Stock
- Units On Order
- Reorder Level
- Discontinued


22. Sample Suppliers (total 38 unique records)

- Company Name
- Contact Name
- Contact Title
- Address
- City

- Region
 - Postal Code
 - Country
 - Phone
 - Fax
 - Home Page
- 23. SIC Codes and Industry Names (total 443 unique records)**
- SIC
 - Industry Name
- 24. Stock Quotes – S&P 500 (total 500 unique records)**
- Symbol
 - Company
 - Country
 - GICS
 - Sector
 - Price
- 25. United Kingdom Cities (total 66 unique records)**
- City Name
- 26. USA Counties (total 3092 unique records)**
- County Name
 - State Name
- 27. USA Holidays by Year (total 78 unique records)**
- Date
 - Holiday
- 28. USA Social Security Numbers (total 500 unique records, not in use SSN)**
- SSN
- 29. USA States (total 500 unique records)**
- 2-character State Code
 - State Name
- 30. USA Zip Codes (total 3000 unique records)**
- Zip
 - State Code
 - City Name
- 31. World Currencies (total 3007 unique records)**
- 3-character Currency Code
 - Country Name
 - Currency Name

How Random Are Generated Random Values?

To ensure that generated random values are not pseudo-random and do not repeat every time you run the load process DB Tools uses time functions to generate initial seed values for the random number generators. The same method is used for both the back-end random number generator when using [FAST SCRIPT](#) loading method and the front-end random number generator when using [DATA LOAD](#) loading method.

 **Note:** All mentioned methods use UNIFORM value distribution.

For your information: NORMAL distribution means that the outputs are distributed over a bell-shaped curve. It usually has a zero average and a standard deviation of 1. NORMAL distribution is a good way of describing the distribution of measurement errors. For example, if 0.5L bottles are filled with water with an error of 0.01L (standard deviation) it can be seen that on average 68% of the bottles are within the margin of error. Distribution is said to be UNIFORM if the probability of generating one value rather than another is the same. This is what happens when throwing a dice. The probability of getting number 2 is the same as for number 6.

DB Benchmark Expert

Overview

DB Benchmark Expert enables you to setup a benchmark test that you can use to measure performance of your database doing on-line transaction processing work. This database benchmark test simulates a real workload by allowing you to establish large number of concurrent connections performing user-defined transactions simultaneously. This way you can simulate real database users running various database transactions at a high rate performance. The benchmark rules do not necessary reflect your database real-world operations, however the benchmark takes into account real transactions made by other users while the test is being performed. DB Benchmark Expert provides you with an option to customize transaction involved in the test. Even if you use default settings, you can still use benchmark results for the performance bottlenecks analysis and other feature-performance considerations.

Because of the number of various interrelated factors the benchmark results for your database may vary at different times. Note that the following factors can significantly affect the results:

1. Server hardware configuration
2. Server Operation System configuration
3. Effective values of database instance parameters.
4. Real-world database workload at the time the benchmark is conducted.
5. Types of transactions and transaction specifics.
6. Number of concurrent connections.
7. Client workstation (benchmark host) performance.
8. Network traffic at the time the benchmark is conducted.

Running DB Benchmark Expert

The DB Benchmark Expert opens one or more simultaneous connections to the database. Each connection is created in a separate thread. You specify number of connections, transaction types, and test interval on the "step 2". All opened connections are used to run simultaneously the specified transactions. The order of transactions is random. You can modify the SQL that will be used for transactions on the "step 3". Make sure you specify valid SQL statements to achieve the correct results. The default benchmark settings and phases are:

1. Create and populate transaction table OM_BENCH.
2. Open 10 connections then use all connections to run simultaneously transactions of SELECT, INSERT, DELETE, and UPDATE types involving transaction table. This phase lasts 60 second.
3. Terminate all previously opened connections then drop transaction table.
4. Evaluate results and build TPS (transactions per second) graph.



Tip: To postpone the benchmark start until later time you can specify the required start time in the "step 2". This allows you to prepare several client computers to start benchmarking at the same time, spread the load and therefore achieve better benchmark test quality eliminating the performance dependency related to the performance of your computer from which you launch the test.

Before You Begin

You must have SELECT privileges on the following two dynamic performance tables: V\$VERSION, V\$SYSSTAT. You must also have CREATE TABLE and DROP TABLE privileges. If you do not have sufficient authority, ask your database administrator to grant you privileges mentioned before.

Before You Run DB Benchmark Expert

Choose SQL statements that you want to use for benchmarking. Best SQL statements for benchmarking are those that generate more transactions rather than return or update more records. Queries from your real applications are good candidates for using in performance testing, especially typical SQL statements that run frequently and generate a moderate amount of logical I/O. Ideally you should run DB Benchmark Expert with these queries periodically, perhaps, every week to monitor and compare changes in your application performance and response times.

After You Ran DB Benchmark Expert

Do not rush to tune your database to improve benchmark results. Tuning a database is sometimes like squeezing a balloon - reducing response time for one query often results in the increase of another. Carefully consider your database goal and utilization before making any changes.

Supported Benchmark Types

The following types of benchmarking and load simulation are supported:

- **Server relative performance (TPS)** - you can use this benchmark type to measure your Oracle database server relative performance and then use it as a benchmark for other Oracle servers. You can use either pre-defined SQL queries to run this type of test or use your own. At the end of the test, DB Benchmark Expert generates a bar graph indicating the achieved TPS (transactions per second) rate. Faster servers typically process more transactions and generate higher TPS numbers.
- **Server performance / load dependency** - you can use this benchmark type to measure your Oracle database server performance dependency for different numbers of concurrent user sessions. Up to 4 pre-defined or user-supplied SQL queries can be used to run this type of test. At the end of the test, DB Benchmark Expert generates a line graph indicating the achieved TPS (transactions per second) rate for different numbers of sessions. The graph can be also used to find out your server optimal throughput.
- **Application performance / load dependency** - you can use this benchmark type to measure your Oracle database server performance dependency for different numbers of concurrent user sessions. This type is similar to the previous type, except that it allows you to specify virtually any number of different queries that you want to use for the testing. It is your responsibility to provide valid queries. At the end of the test, DB Benchmark Expert generates a line graph indicating the achieved TPS (transactions per second) rate for different numbers of sessions. The graph can be also used to find out your server optimal throughput.
- **Application load simulator** – you can use this type for simulation your application running by multiple concurrent user sessions and testing how your database server can deal with multiple users. In other words, you can use this type of testing to make your Oracle server busy running application queries and measure your server average response and transaction processing times. This type also allows you to identify the slowest parts of the application code when running under heavy load and multiple sessions. See the following section for more details

Using DB Benchmark as Application Load Simulation Tool

In order to use DB Benchmark as an Application Load Simulation tool, you should first prepare the application queries to be used for simulation. In case you have access to the application source code, you can copy most important queries from the source. If you don't have access to the source code, you can use a variety of methods to obtain the SQL code. For example, you use the [DB Application Expert](#) module to monitor your applications and to collect their SQL queries. You can then copy and paste these queries to DB Benchmark.

Application Load Simulation is a simple and effective method that can provide meaningful answers to complex questions about your system capacity, multi-user access, application locking and concurrency, and other similar questions:

Typical questions that can be answered with the help of DB Benchmark's Application Load Simulation tool include:

- If transaction volume doubles or triples, can our system handle them?
- How concurrent user sessions affect our application response time?
- Can we safely add 50 more users to an existing server?
- Will the application that we're thinking of acquiring work well for our company?
- A locking problem reported in production environment cannot be easily reproduced in test environment in which a single tester runs the application.
- Which parts of our application perform worse in a multi-user environment and can cause application locking.

Profiling Application Code in Load Simulation Mode

In Application Load Simulation mode, DB Benchmark can internally invoke PL/SQL Profiler for all launched user sessions, and then analyze and aggregate the results.

In order to use the profiler option, you must first setup DB Tools [SQL Profiler](#) module. If you haven't done this already, start the SQL Profiler and following the prompts.

To turn on the PL/SQL profiler in the Application Load Simulation mode, tick "Enable SQL profiling" checkbox in the load test options.

At the end of the test run, the aggregated profile report will be displayed automatically within DB Benchmark main window. The report will indicate top 20 code lines (in other words, operations) by their average execution time. To get more or less meaningful results, it is recommended to use code units for the test that involve execution of named units in the database, such as stored procedure or package calls, trigger invocation, user-defined functions and other user-defined code-based units.



Tip: To review past Load Simulation results and to do detailed and comparative analysis, you can use the DB Tools [SQL Profiler](#) module. DB Benchmark's generated Application Load Simulation tests are saved automatically as SQL Profiler projects having names in the following format.

DB Benchmark Load Simulation [test start date and time]

Such names can be easily recognized in the list of SQL Profiler projects. Each user session used for the test is saved as a separate run with the project. You can analyze individual sessions as well as compare their performance and dependencies at the code unit and code line levels.

DB Zip/Unzip Expert

Overview

Have you ever used ZIP files? If so, you've probably encountered some Zip/Unzip utilities that provide easy-to-use interface to handle these archived ZIP files. However, these utilities cannot handle data stored inside databases. DB Zip/Unzip brings the convenience of these utilities to the world of Oracle databases. DB Zip/Unzip is also a quick and easy way to migrate and backup data stored in your Oracle databases. With a few mouse clicks you can now zip tables, schemas or even entire Oracle databases.

DB Zip/Unzip Expert also complements [DB Copy Expert](#) utility. While DB Copy Expert can be used to copy data between 2 on-line Oracle databases, it cannot be used to unload the data into files, save them on some media and then later restore it to the same or different database. As opposite to DB Copy Expert, DB Zip/Unzip can be used for off-line data unloading and loading as well as for data archival purposes.

DB Zip features include:

- DB Zip captures full table DDL including columns, storage options, constraints, indexes, triggers, grants, and even table and column comments.
- DB Zip is lightning fast, an average 100,000 rows table can be zipped on a Pentium II class computer in just 2 minutes.
- High compression level: many small and medium size schemas can fit on a single floppy disk.
- DB Zip creates industry standard zip files that are compatible with most zip/unzip utilities.
- Zip files containing compressed data allocate much less space than their tables of origin.
- Zipped file can be copied or transferred using FTP or simply emailed to another office.

DB Unzip can be used to restore compressed tables and data. DB Unzip features include:

- Compressed tables can be optionally unzipped to
 1. Another database

or

 2. Recreated in the same schema of the origin database

or

 3. Recreated in another schema of the origin database
- Data from the zipped tables can be optionally
 4. Appended to already existing tables

or

 5. Used to replace contents of existing tables

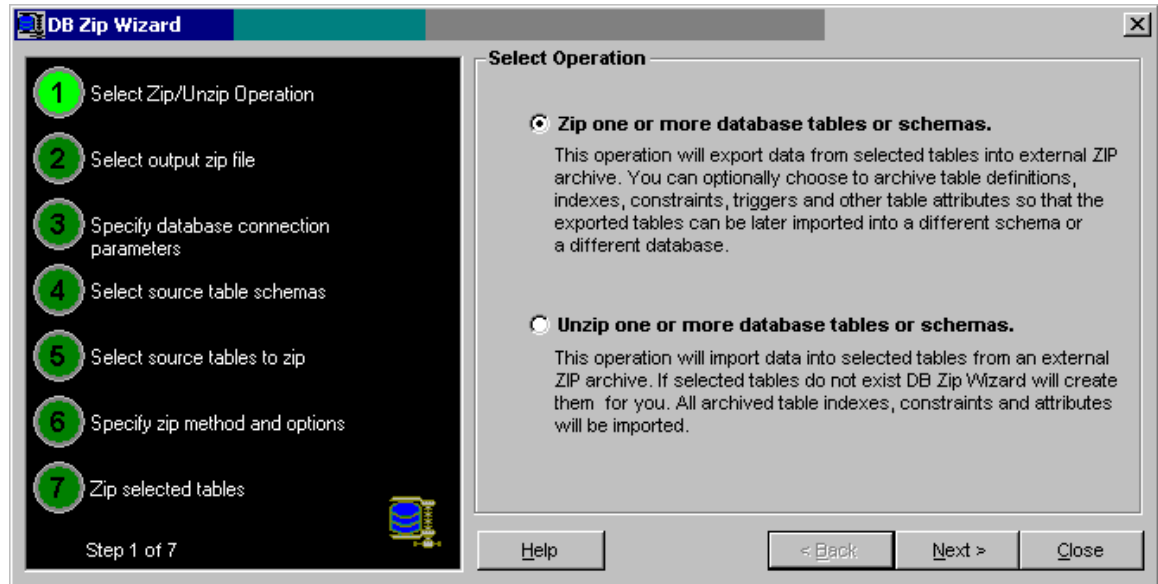
or

 6. Inserted to existing tables having the same or different columns
- Unzipping is as simple as zipping and can be performed with just a few mouse clicks.
- All operations are completely logged.

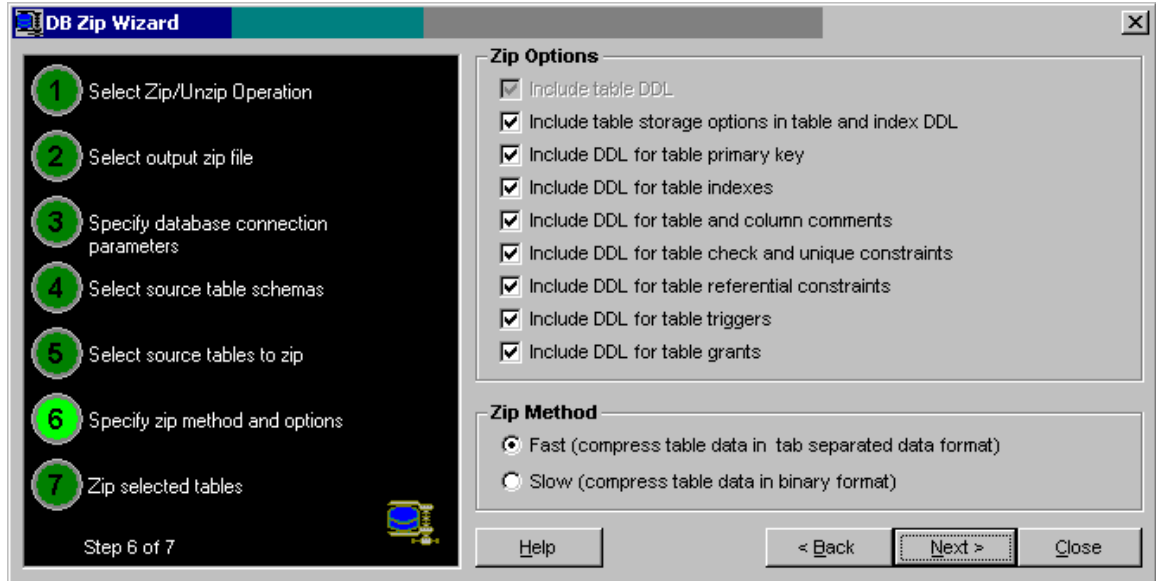
Zippping Tables


Use the following steps to zip one or more database tables:

1. Choose File/Zip-Unzip Wizard menu (Shortcut CTRL E). The DB Zip Wizard dialog will appear



2. Click the **Next** button to advance to the next step.
3. Enter Destination Zip File name. Type name of a new file or select an existing file.
Note: If you select an existing file this file will be overwritten.
4. Specify source database connection parameters and then click the **Next** button. The Wizard will display list of all database schemas.
5. Select one or more schemas containing tables that you want to zip and then click the **Next** button again.
6. Select one or more tables that want to zip. To select multiple tables use common DB Tools keyboard shortcuts involving SHIFT and CTRL keys. Click the **Next** button again
7. Select appropriate zip method and additional zip options such as what parts of the table reverse-engineered DDL script you want to add to the zip file. This DDL can be later used by DB Unzip Wizard to restore the selected tables in the same or another Oracle database. Click the **Next** button again.



 **Important Notes:** DB Zip supports 2 different zip methods:


- "Fast" method uses tab-separated data format. This method is very fast, but it cannot be used for tables containing binary data in text-like columns. If you pick "Fast" method for such tables you will not be able later to unzip the resulting file and restore the original data completely.
- "Slow" method uses binary data conversion and because of that it can be used to zip any data, but it is generally takes more time for data compression and decompression.

8. Select data processing scope options and then click the **Finish** button. For more information about these options see [Using DB Zip/Unzip for Quality Assurance and Development](#) topic. After this step DB Zip will begin to zip selected tables.

Unzipping Tables

Use the following steps to unzip one or more database tables from a zip file previously created using DB Zip Wizard:

1. Choose **File/Zip-Unzip Wizard** menu (Shortcut CTRL E). The DB Zip Wizard dialog will appear.
2. Select the **Unzip** option and click the **Next** button to advance to the next step.
9. Enter Source Zip File name. Type name of an existing file or select an existing file previously created using DB Zip Expert and then click the **Next** button to advance to the next step.
3. Specify source database connection parameters and then click the **Next** button again.
4. Select one or more tables that you want to unzip. If necessary correct their names as you want them to appear in the target database. Click the **Next** button again.

 **Important Note:** Target tables can have names and schemas different from their tables of origin.

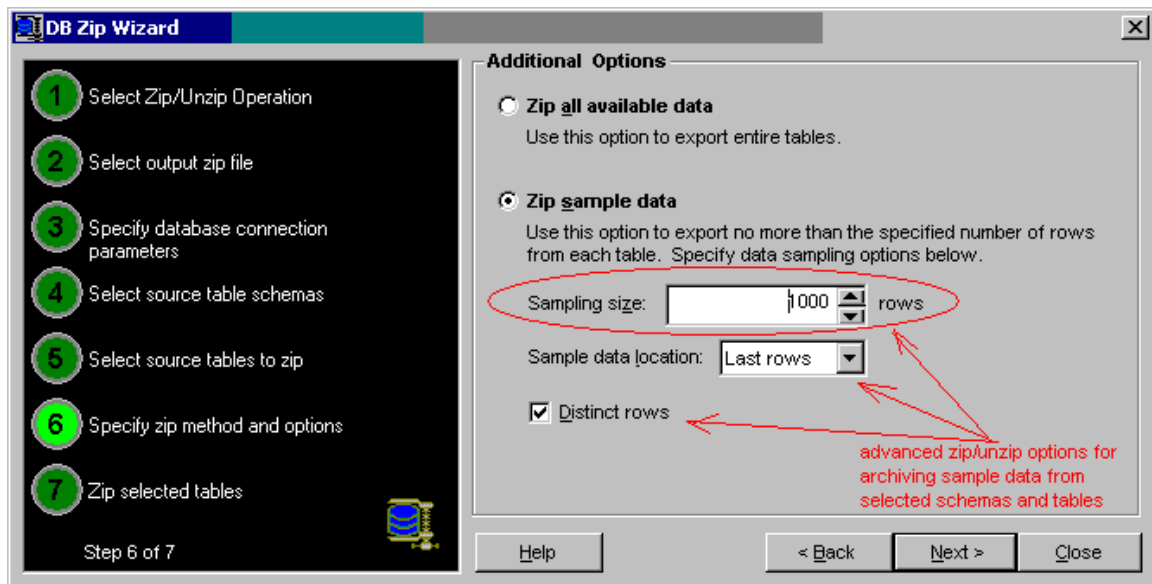
5. Select any additional unzip options such as parts of table DDL you want recreate and how you want to deal with existing tables that have the same name as tables being unzipped.
6. Click the **Next** button again. After this step, DB Unzip will begin to unzip selected tables.

Using DB Zip/Unzip for Quality Assurance and Development

DB Zip/Unzip is a very handy tool in Quality Assurance (QA) and Development environments because it allows partial data export based on the user specified criteria and also because it can process entire schemas at once. Here is why it is an essential to for all QA and developers:

- Using DB Zip/Unzip Expert you can bring small subset of your production data (from your Production database) to your QA or development environment in a matter of minutes.
- Using DB Zip/Unzip Expert you can archive different sets of test and QA data to files and quickly restore them whenever you need to test a different case or review a different scenario.
- Using DB Zip/Unzip Expert you can temporarily archive data to files, then run your development scripts to rebuild your database tables and after that use DB Zip/Unzip Expert again to restore the data lost during table rebuilds or reorganizations.

The following screenshot demonstrates how to use advanced DB Zip options to archive data samples.



DB Copy Expert

Overview

DB Copy Expert is a must-have tool for every Oracle DBA and developer who works with multiple Oracle databases. DB Copy Expert can be used in the following scenarios:

- Copy one or multiple objects (Views, Procedures, Triggers, Indexes, etc...) from one database to another.
- Copy tables with and without data from one database to another.
- Copy data samples – best for copying samples of production data to development or QA environment.
- Copy entire schema from one database to another. DB Copy Expert intelligently resolves schema object dependencies and referential constraints and ensures that schema objects are copied in the proper order.
- Copy objects between different database versions. Copied objects are automatically downgraded or upgraded depending on the version of the target database.

Best of all DB Copy Expert features intuitive graphical interface allowing to copy objects between databases using convenient drag-and-drop methods that make copying objects and data between databases as simple as copying files between folders using Windows Explorer.



Important Notes:


- In order to copy objects and data you must have sufficient privileges in both databases.
- Code-based objects (such as procedures, triggers, etc...) copied to another schema get compiled in the target database using privileges of the user performing the copy process rather than the target schema owner.
- The copy operation will completely fail if you choose "copy storage settings" option for database tables and the target tablespace does not exist or the owner of the target schema does not have sufficient space quota for the target tablespace.
- You must have GRANT ANY privilege in order to copy object grants for objects in other database schemas. Please note that by default this privilege is not given to anybody even not to the DBA role. To obtain this privilege login to the database using the SYS user and grant this privilege to yourself. You can use [DB Consumer Expert](#) or [DB Security Expert](#) to modify user privileges graphically or you can simply execute `GRANT GRANT ANY TO <user>;` command in DB Tools [SQL Editor and Debugger](#).
- When copying database schemas DB Copy Expert attempts to resolve schema object dependencies and referential constraints and copy objects in the proper order. However the copy operation may fail if any object being copied references objects in some other schemas or has circular dependencies.

Copying Objects and Schemas

Getting started

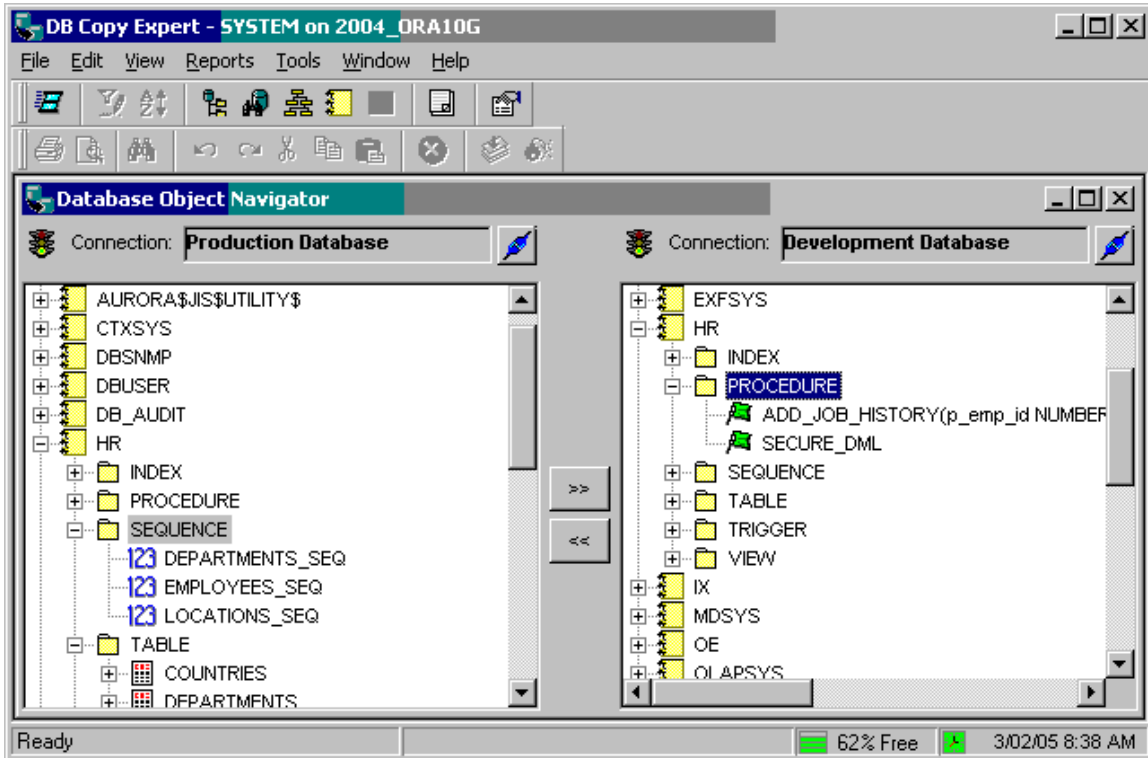
The first thing you should do after starting DB Copy Expert is to connect to the source and target database systems. Until you connect to at least one database nothing is enabled in the DB Copy Expert graphical interface.


On startup DB Copy Expert automatically opens the Database Object Navigator window. This window features 2 panes - one for each database. It does not matter which pane you use for the source database and which pane you use for target database. Objects can be copied both from left to right and from right to

left. Use the Connect  button displayed on top of each pane to connect to your database system. Clicking on that button brings up DB Tools standard [Connect To Database](#) dialog. Pick different databases



for each pane connection. If connection is successful DB Tools populates the active pane with name of database schemas and their objects.

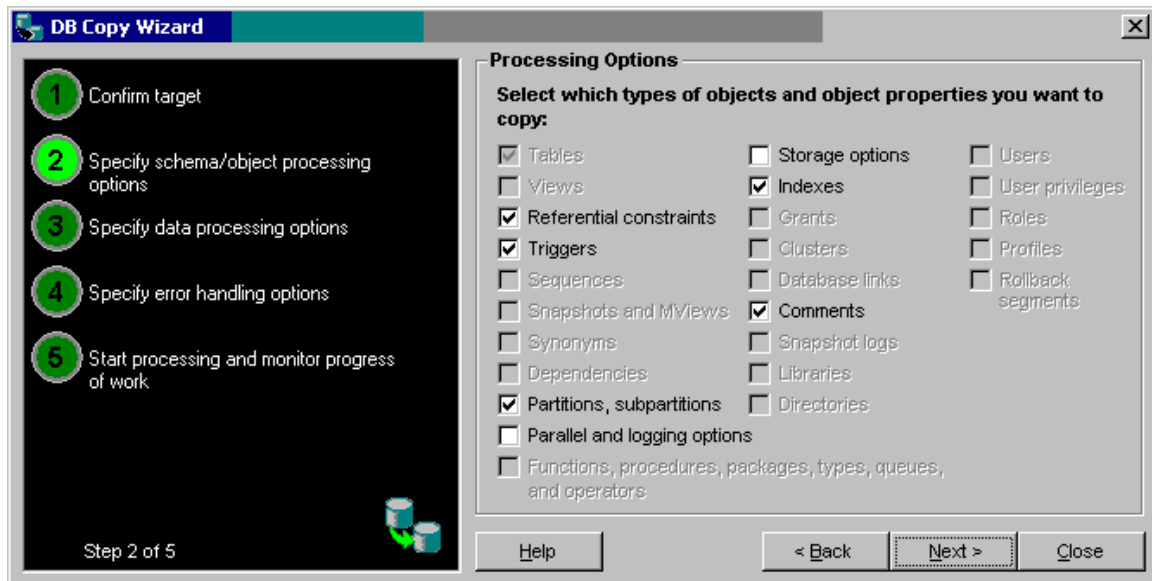
Sample Database Object Navigator screen follows.



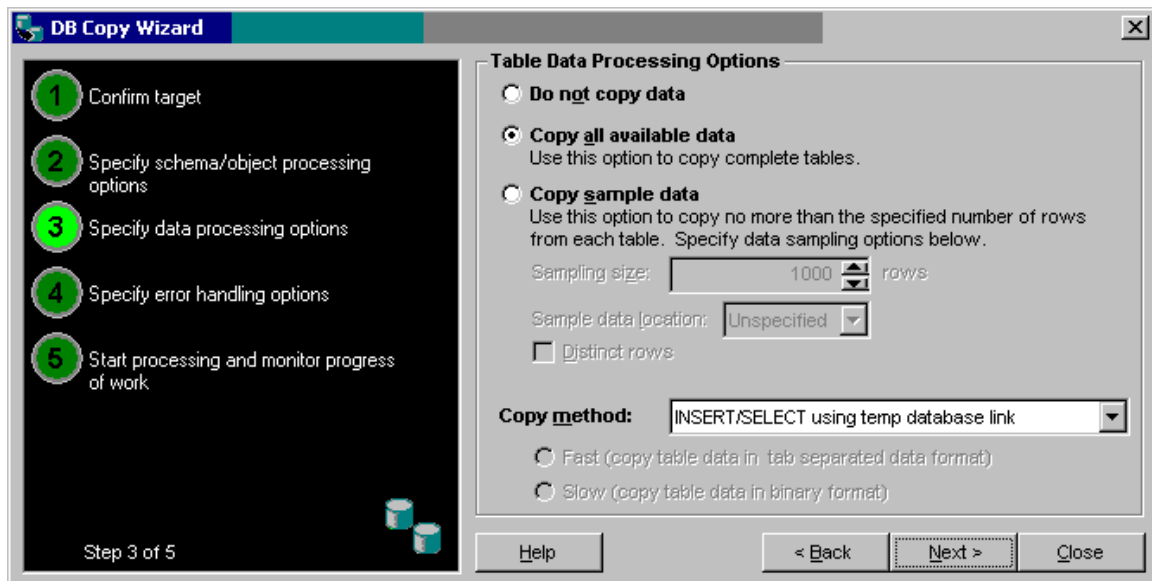
 **Tip:** Clicking on any schema or object in the left or right pane makes that pane and the associated database connection "active". The name of the database server for the active connection is always displayed in the DB Copy Expert title bar. All reports and tools you run using DB Copy Expert menus utilize the active connection as their data source.

Copying database tables

1. Open Database Object Navigator screen if it is not opened already.
2. Connect to your databases using the Connect  buttons. For example, use the left side of the screen to connect to your "source" database and right side of the screen to connect to your "target" database.
3. On the left side of the screen ("source") locate the required schema in the database tree-view and expand that schema folder
4. If you are copying objects to an existing schema in the target database then you must select that schema on the right side ("target") of the screen. If you are copying objects belonging to a schema, which does not exist in the target database, and you want DB Tools to create that schema for you then on the right side of the screen you must select the top item in the tree . This top item with the database server name.
5. On the left side of the screen expand the TABLE folder and select the table you want to copy.
6. Use either drag-and-drop or click the "copy to right" >> button to copy the selected table from the left side of the screen to the selected schema (or database) on the right side of the screen. In response to this action the DB Copy Wizard dialog will appear.



7. Confirm the copy operation by clicking the **Next** button. Check all options that you want to copy with the table and then click the **Next** button again to advance to the next step.
8. Select data processing options and method. If you don't want to copy table data check **Do not copy data** option. When ready click the **Next** button again to advance to the next step.



 **Tip:** DB Copy Expert supports 2 different copy methods:

- **INSERT/SELECT using temp database link** – If this method is chosen DB Copy Expert will create a temporary database link in the source database and then use that link to run `INSERT INTO <target table>@<link name> SELECT * FROM <source table>` command in the source database. This method that it can be used for both regular tables and object tables. **The disadvantage of this method is that it requires a temporary database link created for the duration of the copy operation. This temporary link is not a "temporary" object from a database point of view and during the copy operation other people can potentially use it and gain temporary access to the target database.** The link is automatically dropped after the copy operation is complete.

- **Export/import using external file** – If this method is chosen DB Copy Expert internally uses [DB Zip Expert](#) to export source table data to a temporary file and then import it to the target table. This method can be used for regular tables only. **The disadvantage of this method is that it requires an temporary work-file created on your local disk for the data export/import operation. If you are copying a lot of data using this method, make sure you have sufficient free disk space to store the file.** The file is automatically deleted after the copy operation is complete.



Tip: For more information about supported data copy methods using export/import option and details on copying data samplings, see [Using DB Zip/Unzip for Quality Assurance and Development](#) topic.

9. Specify appropriate error handling options and click the **Next** button to begin the copy process.





Important Notes:

If while copying a partitioned table you don't select the **Copy partitions and subpartitions** option DB Copy Expert will then create the target table as a non-partitioned table and then attempt to copy data from all partitions and subpartitions of the source table.


Also note that with partitioned tables you should not use the data sampling options for copying "First" and "Last" records because these options will not work correctly as there are different sets of first and last records for each partition and subpartition and the data copy operation is applied to the entire table.

Copying other objects

1. Open Database Object Navigator screen if it is not opened already.
2. Connect to your databases using the Connect  buttons. For example, use the left side of the screen to connect to your "source" database and right side of the screen to connect to your "target" database.
3. On the left side of the screen ("source") locate the required schema in the database tree-view and expand that schema folder
4. If you are copying objects to an existing schema in the target database then you must select that schema on the right side ("target") of the screen. If you are copying objects belonging to a schema, which does not exist in the target database, and you want DB Tools to create that schema for you then on the right side of the screen you must select the top item in the tree . This top item with the database server name.
5. On the left side of the screen expand the appropriate folder and select the object you want to copy.
6. Use either drag-and-drop or click the "copy to right" >> button to copy the selected object from the left side of the screen to the selected schema (or database) on the right side of the screen. In response to this action the DB Copy Wizard dialog will appear.
7. Confirm the copy operation by clicking the **Next** button. Check all options that you want to copy with the selected object and then click the **Next** button again to advance to the next step.
8. Specify appropriate error handling options and click the **Next** button to begin the copy process.


Copying groups of objects

DB Copy Expert can copy multiple objects of the same type in a single copy operation. Essentially the processing in this case is the same as for a single object and requires exactly the same steps, which are described in detail in the [Copying database tables](#) topic. The only difference is the step 5. On this step you must select the entire folder for the object group (for example, TABLE folder or PROCEDURE folder) rather than an individual object.

 **Tip:** When copying groups of database objects DB Copy Expert attempts to resolve object dependencies and referential constraints and copy objects in the proper order. However the copy operation may fail if any object being copied references other objects that do not exist in the target database or has circular dependencies.

Copying entire schemas

DB Copy Expert can copy an entire schema at once. Essentially the processing in this case is the same as for a single object and requires exactly the same steps, which are described in detail in the [Copying database tables](#) topic. The only difference is the step 5, which is not required when copying entire schemas.

 **Tip:** When copying entire schemas DB Copy Expert attempts to resolve object dependencies and referential constraints and copy objects in the proper order. However the copy operation may fail if any object being copied references other objects in other schemas that do not exist in the target database or has circular dependencies.

Reports

For your convenience DB Copy Expert provides a number of object summary and detail reports organized by object type. Before running a report make sure to activate proper pane in the Database Object Navigator. Depending on your selection DB Copy Expert will run reports using either "source" or "target" database connection. The name of the database server associated with currently active connection is displayed in the title bar of the main DB Copy Expert frame window.

Besides simple report viewing and printing, you can export report data to numerous file formats as well as select and copy report data to the Windows clipboard for sharing with other applications. See [Working with data and export](#) topic for help on how to export, copy, filter and sort report data. You can also right-click on any report to get some additional features accessible from a popup menu that appears in response to the click.

For most reports, DB Monitor Expert displays [Select Schema dialog](#) before retrieving the report. You can check either **Report All** checkbox for all schemas, or select one or more schemas you want to add to the chosen reports.

The following reports are available:

Objects Report

Reports summary data for different types of database objects.

Tables Report

Reports detailed information about all database tables. This report queries database repository table - SYS.DBA_TABLES. Oracle updates table statistic in the repository when you [analyze tables](#). To refresh table statistic use [DB Monitor Expert](#) component. Select **Tools/Table Analyzer** in that component and perform analyzes for tables that haven't been analyzed for a while.

Indexes Report

Reports detailed information about all database indexes. This report queries database repository table - SYS.DBA_INDEXES. Oracle updates table statistic in the repository when you [analyze indexes](#). To refresh index statistic use [DB Monitor Expert](#) component. Select **Tools/Index Analyzer** menu in that component and perform analyzes for indexes that haven't been analyzed for a while. Alternatively, you can analyze tables and/or clusters with cascade indexes option on.

Clusters Report

Reports detailed information about all database clusters. This report queries database repository table - SYS.DBA_CLUSTERS. Oracle updates cluster statistic in the repository when you [analyze cluster](#). To refresh cluster statistic use [DB Monitor Expert](#) component. Select **Tools/Cluster Analyzer** in that component and perform analyzes for clusters that haven't been analyzed for a while.

Tablespaces Report

Reports space allocation information for tablespaces; displays free space in red for tablespaces that have less than 10% of free space. A bar graph is provided for easy comparison of free and allocated space.

Database Files Report

Reports database files, their sizes and statuses; for each file reports tablespaces, it consists of, as well as tablespace sizes and statuses.

Sequences Report

Reports detailed information about all database sequences.

Snapshots Report

Reports detailed information about all database snapshots, including query text for each snapshot.

Synonyms Report

Reports detailed information about all database synonyms.

Database Links Report

Reports detailed information about all database links.

Users Report

Reports detailed information about all database users, including encrypted password. This allows DBA and application developers temporarily replace passwords for certain accounts, in order to test functionality of the database applications.

Table Privileges Report

Reports all granted table privileges. For your convenience, the report is broken on four groups:

- Group by grantor
- Subgroup by grantee
- Subgroup by scheme
- Subgroup by object

Triggers Report

Reports detailed information about all database triggers, including trigger definitions.

Procedures, Functions And Packages Report

Reports detailed information about all database function and packages, including their code.

Views Report

Reports detailed information about all database views, including view definitions.

Parameters Report

Reports parameters of the running Oracle instance.

Version Report

Reports versions of the running Oracle server modules.


Custom Reports

See [Custom Reports](#) topic in [DB Monitor Expert](#) chapter for more information on how to create user-defined reports.

Additional Tools

Analyzing object dependencies


For your convenience DB Copy Expert provides direct link to the Object Dependencies Analyzer utility, which is also available in [DB Monitor Expert](#) component. To access this utility click

Tools/Analyzers/Dependencies Analyzer menu or click the Dependencies Analyzer  button on the application toolbar.

See [Dependencies Analyzer](#) topic in DB Monitor Expert chapter for detailed description of this utility and usage instructions.


Searching objects in the database

For your convenience DB Copy Expert provides direct link to the Database Object Finder utility, which is also available in DB Tools SQL Editor and Debugger and DB Monitor components. To access this utility click

Tools/Find in Database menu or click the DB Find  button on the application toolbar.

See [Database Object Finder](#) topic in SQL Editor and Debugger chapter for detailed description of this utility and usage instructions.

Finding invalid objects

For your convenience DB Copy Expert provides direct link to the Status Analyzer utility, which is also available in [DB Monitor Expert](#) component. To access this utility click **Tools/Analyzers/Status Analyzer** menu or click the Status Analyzer  button on the application toolbar.

See [Status Analyzer](#) topic in DB Monitor Expert chapter for detailed description of this utility and usage instructions.

Recompiling invalid objects

DB Copy Expert supports 2 different methods for recompiling invalid objects:

Recompile invalid objects using Database Object Navigator screen


1. Select invalid object in the source or target databases tree-view

2. Right-click the selected object name. In the popup context menu select **Recompile/Selected Object** command
or
Select **File/Recompile/Invalid Object** command in the DB Copy Expert main menu.

To recompile all invalid objects in the same folder or recompile everything in the same folder use the same two step procedure but select either **All Invalid Object** or **All Objects** command in the popup or main menus.


Recompile invalid objects using Status Analyzer utility

1. Start the Status Analyzer utility and let it find all invalid objects in the database.
2. Use methods described in the previous topics to recompile invalid objects. For detailed description of this utility and usage instructions see [Status Analyzer](#) topic in DB Monitor Expert chapter.

 **Tip:** Using Status Analyzer utility method has an advantage over Database Object Navigator method because the Status Analyzer can find all invalid objects in the database and present them all together on a single screen while with Database Object Navigator you have to locate invalid objects yourself and recompile them either one-by-one or group-by-group, schema-by-schema.

Schema reverse-engineering

For your convenience DB Copy Expert provides direct link to the Schema Reverse-Engineering utility, which is also available in DB Tools SQL Editor and Debugger component. To access this utility click

Tools/Schema Reverse-engineer menu or click the Schema  button on the application toolbar.

See [Schema Reverse-engineering](#) topic in SQL Editor and Debugger chapter for detailed description of this utility and usage instructions.

SQL Editor and Debugger

Overview


Here in this topic and in the following topics DB Tools' SQL Editor and Debugger is called for simplify DBA Notepad.

DBA Notepad is a full-featured SQL and PL/SQL editor, which you can use to control database access, create and modify users and groups, create, import, paste, execute, test, debug and tune SQL statements and PL/SQL blocks.

DBA Notepad can execute 4 different types of commands:

- Oracle standard SQL commands
- Oracle standard PL/SQL blocks
- Many Oracle SQL*Plus® compatible commands, for commenting code, formatting query results and setting editor's options
- DB Tools SQL extensions such as CONNECT, DEFINE and other commands for controlling SQL script execution flow

Creating new SQL script

To create a script select **File/New** command (shortcut: CTRL N) or simply click  icon on the program [toolbar](#).

This will open a new DBA Notepad window where you can start typing your SQL script.

Opening previously saved SQL script

You can open an existing script file into DBA Notepad window by doing any of the following:

Select **File/Open** command (shortcut: CTRL O) or click  icon on the program [toolbar](#).

After you open a script file, you can start editing it by typing changes or immediately execute SQL statements from it.



Tips:

- You can also use last four items from the **File** menu to open four most recently edited files.
- You can also double-click SQL file in the [SQL Navigator](#) to open that file in the DBA Notepad.
- If you start DB Monitor Expert from the command line with the SQL file as a parameter, it will open the specified file in the DBA Notepad

Using drag-and drop

You can also open files in DBA Notepad by dragging their icons from Windows Explorer and dropping them onto SQL Editor/Debugger or DB Monitor Expert windows. DB Tools will automatically create a new DBA Notepad window with the dragged file automatically opened in it.

Importing text of an existing database stored procedure, function package, package body, type, type body or trigger

To import an existing object definition into DBA Notepad:

1. Select **Object/Import SQL** command (shortcut: ALT O, I). DB Monitor Expert will retrieve and display list of all stored procedures, functions, packages, and triggers.
2. Select an object name you wish to import into editor.

After you import SQL, you can start editing it by typing changes or immediately execute SQL statements from it.

Comments

You can use comments to document your SQL scripts and prevent statements within a script from executing. There are two methods.

1. Double-dash `--` method: Designate all text on the line to the right of the double-dash marker as a comment. Such comments cannot extend to multiple lines.

Example:

```
Code – Comment on the same line
– Comment on a separate line
```

2. Slash-and-asterisk `/*...*/` method: Designate the text between the markers as a comment. Such comments can extend over multiple lines (multi-line comments do not require a continuation character)

Example:

```
Code /* Comment on the same line */
/* Comment on a separate line */
```

You also can use block comments with the plus sign to specify execution **hints**. For example: `/*+ RULE */` For detailed information on the specific hints, consult the Oracle documentation.



Note: Comments can appear in any part of the SQL script. However, when parsing and sending SQL to the database, DB Notepad skips comments outside of SQL statements and [PL/SQL blocks](#). Comments that appear inside SQL statements and [PL/SQL blocks](#) are sent to the database as part of the statement.

Variables

DBA Notepad supports substitution variables. A substitution variable is a user-defined variable preceded by the ampersand (&) symbol. When DBA Notepad encounters a substitution variable in a command, DBA Notepad executes the command as though it contained the value of the substitution variable, rather than the variable itself. You use **DEFINE** statement to declare, set initial values and change existing variables. You can also use **View/Show Variables** command to enter and modify substitution variables. You can use **UNDEFINE** statement to remove declaration of a previously defined variable.

DEFINE command syntax:

```
DEFINE <variable name> [value];
```

Value is an optional parameter. If a value is not specified, the specified variable will be removed from the scope.

Substitution variables can hold any text including parts of SQL statements. Variable names are case insensitive.

Examples:

```

DEFINE Schema          'DEMO' ;
DEFINE WhereClause     owner = 'DEMO' AND table_name like 'A%';
SELECT count(*) FROM all_tables WHERE owner = &Schema AND table_name like 'A%';
SELECT count(*) FROM all_tables WHERE &WhereClause;

```

Edit commands, search and replace commands, go to commands

As a full-featured editor, DBA Notepad provides all the common methods for text editing. All these methods are standard for Windows application. For supported hot keys see [hot keys](#) topic.

Supported SQL*Plus commands



The following table describes which SQL*Plus commands and their shortcuts which are currently supported by DBA Notepad.






Command	Description
DOC DOCUMENT REM REMARK	Begins a comment in a command file.
DEFINE	Specifies a substitution variable and assigns it a value. If the variable is already defined this command updates variable's value.
UNDEFINE	Removes previous defined substitution variable
EXECUTE	Executes a stored procedure. The procedure name and parameters must be specified on the same line.
CALL	Executes a stored procedure. The procedure name and parameters could spawn multiple lines.
PAUSE	Pauses script executions and display interactive Continue or Abort message prompt. Clicking the Continue button causes the paused script to resume. Clicking the Abort button aborts the execution.
TIMING	Shows/hides SQL trace window containing timing stats for executing SQL commands.
SHOW ERRORS SHOW ERR	Displays compilation errors pane. The command is ignored if the Errors pane is already displayed.
CONNECT	<p>Connects a given username to the database. The command must be followed by one of the following:</p> <ul style="list-style-type: none"> <username>/<password> – reconnect to the last connected database with a different user name and password. <username>/<password>@<profile name> – reconnect to the database specified in the profile value. <p>Note: This command is especially handy after receiving ORA-03113: end-of-file on communication channel error when the Oracle client to server communication breaks on the server side. Executing this command restores the connection without resetting open screens and dialogs.</p>
DISCONNECT	<p>Closes active database connection.</p> <p>It is not necessary to call DISCONNECT before CONNECT. The CONNECT command automatically closes active connection before attempting to reconnect to the database.</p>
DES DESC DESCRIBE	<p>Describes definition of a table, view, stored procedure, function, package, type or synonym and copies that text to the system clipboard. The copied text can be then pasted to the same Editor window or anywhere else.</p> <p>This command is an extended version of its SQL*Plus sister.</p>

	<p>DESCRIBE uses the following search order when evaluating names specified on the command line:</p> <ol style="list-style-type: none"> 1. Objects in your schema 2. Synonyms in your schema 3. Public synonyms 4. Objects in the SYS schema 5. Objects in the SYSTEM schema <p>For example, executing DESC employees command while connected as DEMO user will copy column definitions of the DEMO.EMPLOYEE.</p> <p>Executing DESC all_tables command will copy definition of columns in the SYS.ALL_TABLES system view.</p> <p>Executing DESC dbms_output command will copy names of public procedures and their parameters in the SYS.DBMS_OUTPUT package.</p>
--	--

Formatting commands

Use **Format/Comment** and **Format/Uncomment** to quickly comment and uncomment the selected text.

Alternatively, you can click  and  icons on the program [toolbar](#). If no text is selected then the current line is processed. The **comment** function comments the selected block of text by adding "--" symbols before each line. The **uncomment** function uncomments the selected block of text by removing "--" symbols from the beginning of each line.

- 1 Use **Format/Upper Case** and **Format/Lower Case** to change case of the selected text. Alternatively, you can click  and  icons on the program [toolbar](#).
- 2 Use **Format/Indent** and **Format/Unindent** to change indenting of the selected text. When you increase or decrease indenting, you change the distance of the text from the margins. Alternatively, you can click  and  icons on the program [toolbar](#).
- 3 Use the **Format/AutoFormat** command to let DB Monitor Expert reformat your script. DB Monitor Expert will find and convert all Oracle reserved words (displayed in blue) to upper case, all other text (displayed in black) will be converted to lower case. DB Monitor Expert will not change strings enclosed in single quotes and comments (displayed in green). Alternatively, you can click  icon on the program [toolbar](#).

Entering SQL

You can enter SQL statements using 5 different ways:

- Pasting statements from the clipboard
- Typing statements in the editor's workspace
- Opening a file containing the SQL using **File/Open** command
- Opening file from SQL File Navigator using **File/Open in DBA Notepad** command
- Importing source of existing code object using **View/Import SQL** command

Statement terminator

SQL Editor uses the semicolon as the SQL statement terminator. SQL Editor automatically recognizes PL/SQL blocks that can include semicolons within the block.

PL/SQL blocks support

SQL Editor parses and automatically recognizes individual SQL statements and PL/SQL blocks containing multiple PL/SQL statements so that no special coding is needed to separate individual statements and PL/SQL blocks. However, for compatibility with previous versions of DB Tools and interoperability with SQL*Plus it also supports using optional braces `}` and forward-slash `(/)` characters for designating multi-statement blocks and executing SQL batches.

To declare a stored procedure, function, other PL/SQL objects or simply execute generic DECLARE...BEGIN...END block, optionally enclose group of related SQL statements in braces `}`. Braces designate beginning and ending of a multi-statement PL/SQL, and ensure it is sent to the database as a single executable command. Please note that a PL/SQL block cannot produce a result set. Use this syntax mostly to execute various forms of Data Definition Language (DDL).

Example:


```
{
    CREATE OR REPLACE FUNCTION db_alert( severity VARCHAR2, message VARCHAR2
    )
    return NUMBER
    IS
    -- severity must be one of the following: 'INFO', 'WARNING', 'ERROR'
    -- message can be any string up to 2000 characters long

    -- return code is the value returned by dbms_pipe.send_message() function
    -- for supported return codes please see Oracle PL/SQL Guide for
    send_message
    BEGIN
        dbms_pipe.pack_message(severity);
        dbms_pipe.pack_message(message);

        RETURN dbms_pipe.send_message('DB_ALERT');
    END;
}
```

Visual Explain Plan

Sometimes there is more than one way to code SQL statements to obtain the required results. When this is the case, you can use **Explain Plan** on the **File** menu to help you select the most efficient method.

[Enhanced Explain SQL](#) window displays information about the path that the database will use to execute the statements. This is most useful when you are retrieving or updating data in an indexed column or using multiple tables and you want tune offending statements. Type SQL statement for which you want to get explanation of the execution strategy and select **File/Explain Plan** command or simply click Plan button on the [toolbar](#). 

Notes:

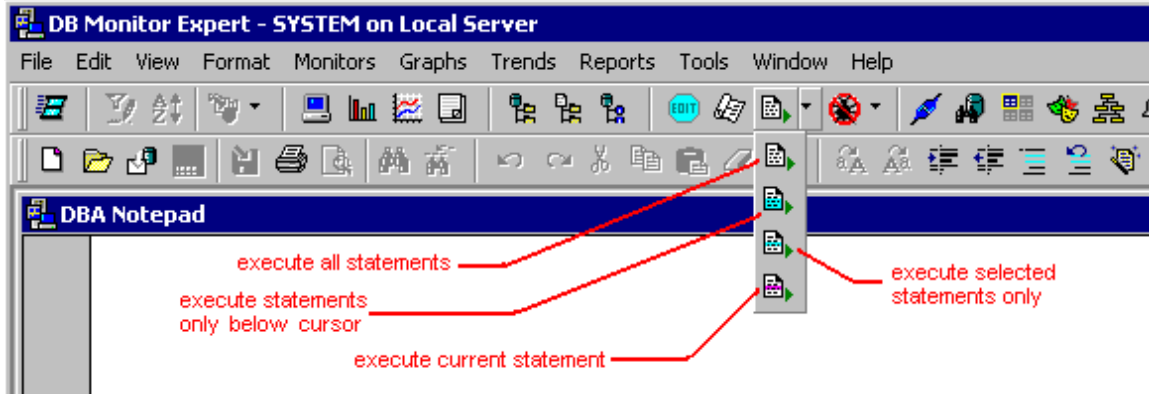
- The Explain Plan window automatically generates plans for your original SQL query and for alternative variants of your SQL query, thus saving your time and presenting you with a choice of execution plans. If you would like to compare generated plans and produced execution characteristics, switch to the **Compare** tab page the click there the **Execute and Compare** button
- In case if you don't want automatic alternative plan generation and collection of extended statistics for objects that are referenced in the plan, you can turn corresponding options in the **Tools/Options**.

Executing SQL

When you have the SQL statements you want in the workspace, you can submit them to the database.

- To execute all statements in the workspace, use **File/Execute SQL/Execute All** command from the menu (shortcut: CTRL R). DBA Notepad parses SQL in the workspace and sends it to the database statement by statement in the order as it is entered in the workspace.
- To execute highlighted statement(s), use **File/Execute SQL/Execute Selection** command (shortcut: CTRL SHIFT+R)
- To execute all statement below the cursor use **File/Execute SQL/Execute After Cursor** command
- To execute the current statement where the cursor is positioned, use **File/Execute SQL/Execute statement below the cursor** use **File/Execute SQL/Execute Current Statement** command (shortcut: CTRL J) or press this button

You can also use toolbar buttons that provide quick access to all supported Execute commands.



If the SQL retrieves data, the data appears in a window identical to a grid Data Manipulation view. For the description of supported data-manipulation and import/export features, see [data-manipulation and data export](#) topic.

If a database error occurs while the SQL is executed, DBA Notepad displays an error message box describing the problem. If the executed SQL contains one or more DDL statements for creating or altering various code objects and Oracle returns compilation errors, such error messages are displayed in the scrollable box on the bottom of the DBA Notepad window. The error message text is displayed along with the line and column of the workspace where the error is detected. If you click on the error message, DBA Notepads automatically moves the cursor to the specified line and column.

1 Use File/Execute

SQL Editor parses SQL in the workspace and submits it to the database statement by statement

Choose **Execute SQL from Caret** command (under the **File** menu) to execute only SQL statements appearing below the editor's caret.

You can also use at (@) command to run the specified command file as a single PL/SQL block.

Syntax: @ file_name

DBA Notepad first will search for the specified file in the current directory. If the file cannot be found in the current directory, DBA Notepad will search a system-dependent path to find the file. If the specified file contains one or more SELECT commands DBA Notepad will execute them but the returned result will be ignored.

Tracing SQL and collecting performance statistics

To start collecting performance statistics for queries executed in the DBA Notepad:

- 1 Click the **View** menu then click the **SQL Trace** command. The SQL Trace window appears

- Execute one or more required queries in the DBA Notepad. SQL Trace captures all executed queries and collects Oracle performance statistics for each query. Statistics for any new queries executed in DBA Notepad are added at the end of SQL Trace window. Use window scrollbars to review all captured SQL.
- Use Print, Filter, Export, Sort menus to print, export, and sort statistics in the SQL Trace window

Debugging SQL

Both SQL Editor and DB Monitor Expert support two debugging methods:

- [Debugging using DBMS_OUTPUT package](#)
- [Debugging using built-in PL/SQL Debugger](#), which provides easy-to-use graphical interface to the Oracle database Probe API.

Database Object Navigator

DB Tools' Database Object Navigator is a comprehensive utility for exploring Oracle databases. Using that utility, you can browse all database schema objects and their properties. You can also browse database tablespaces, redo logs, rollback segments, users, profiles and many other objects and categories.

The Object Navigator window is divided into two parts:

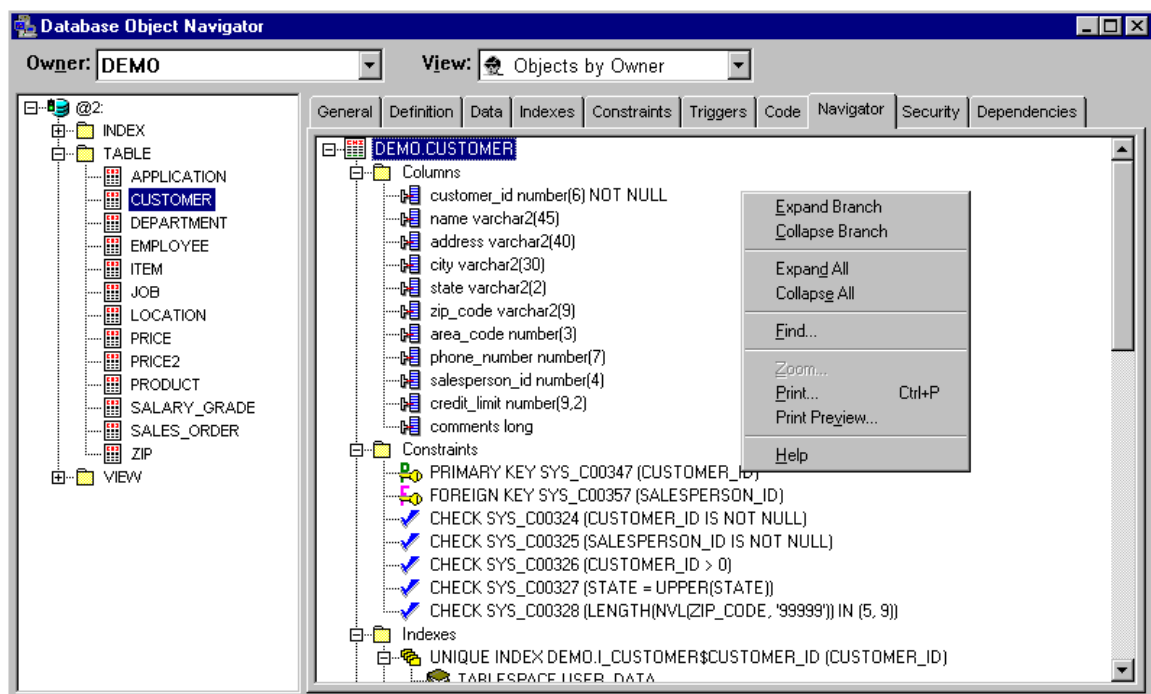
- The "Navigator" - left pane
- The "Object Properties" - right pane

You can change the size of the panes. When you make any pane larger, the adjacent pane becomes smaller. To resize panes:

- Move the mouse pointer in between of the panes until you see a black line with two arrows.



- Drag the edge of the pane until the pane is the size you want.



Views

Database Object Navigator provides four different views of your database:

- View Objects by Owner
- View Objects by Object Type
- View All Objects
- View Database Structure

Use the **View** drop-down list box to select the view you want to use for the database browsing.

Navigating

To navigate the Database Object Navigator use standard tree-view navigation techniques that you use when working with Windows Explorer and other applications. For quick expanding or collapsing of multiple items use right mouse button clicks to invoke the popup menu. Use the following commands available in the popup menu to quickly expand or collapse multiple items:

- **Expand Branch**
- **Collapse Branch**
- **Expand All**
- **Collapse All**

Which items are displayed?

Object Navigator displays only context sensitive items. For example, if there is at least one stored procedure in the expanding schema then Object Navigator will display **PROCEDURE** folder. On expanding of the **PROCEDURE** folder, Object Navigator will display all stored procedures for the selected schema.

Which items are not displayed?

Object Navigator displays only context sensitive items. For example if the expanding schema has no stored procedures, then Object Navigator will not display the **PROCEDURE** folder.

How to select another schema?

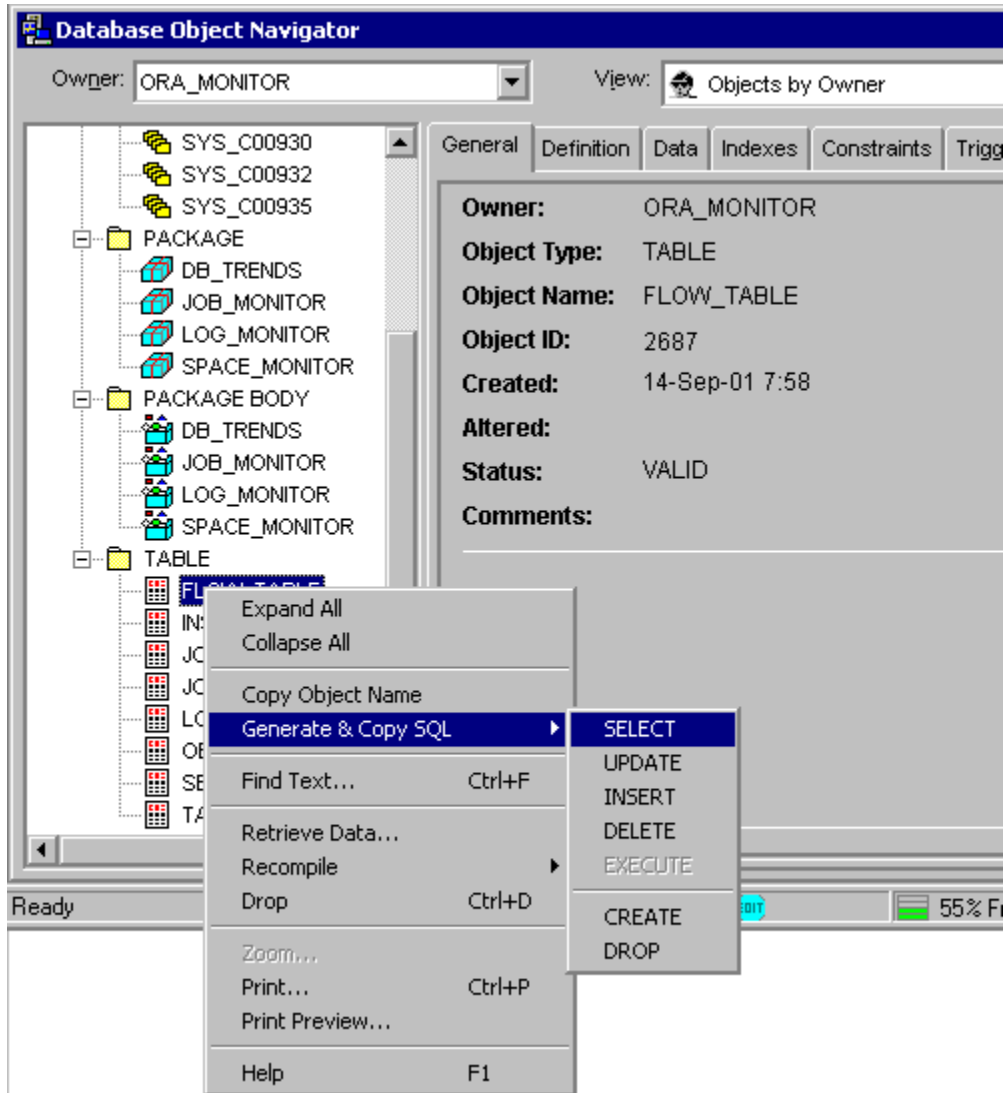
Click on the **Owner** drop-down list on top of the Object Navigator then select the required schema.

How to select another object type?

Click on the **Type** drop-down list on top of the Object Navigator then select the required object type.

How to get generate-common SQL commands for selected database object?

Right-click the object then select Generate and Copy SQL from the popup menu and then select the required command. The Navigator will generate script template for the selected command and paste the generated script into the system clipboard. You can then paste that text into DBA Notepad or other editor.

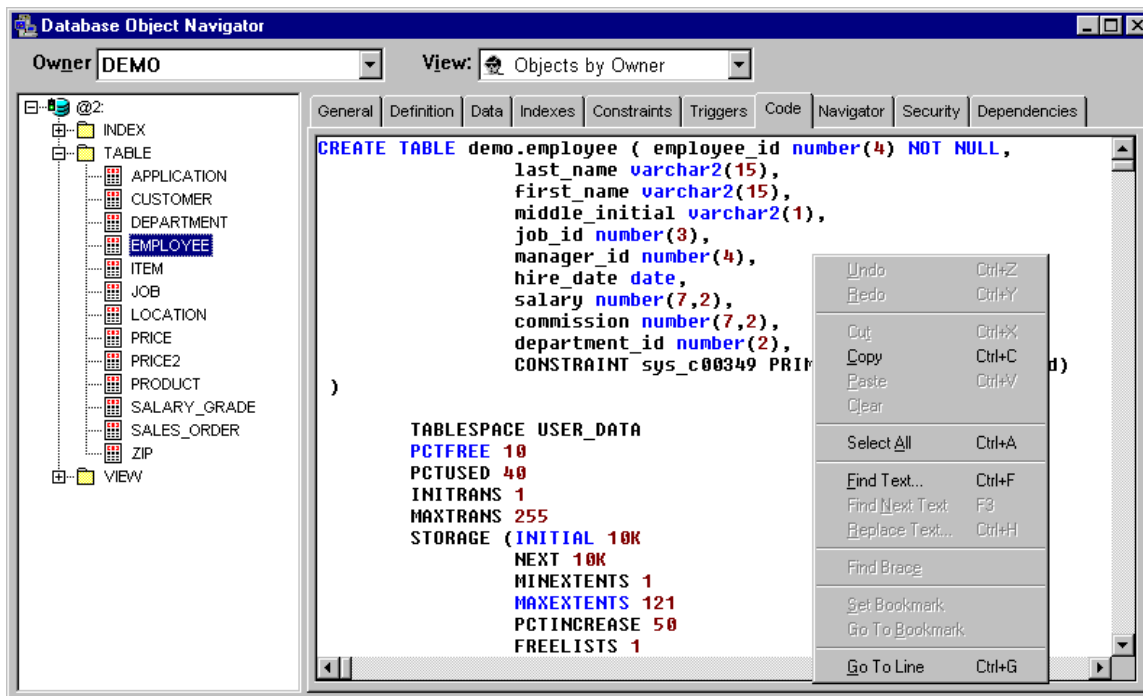


How to reverse-engineer scripts for selected database object?

Click on the **Code** tab of the Object Properties pane.

How to edit procedure, function, package, and table definition code?

1. Click on the Code tab on the Object Properties pane.
2. Right-click the Code Editor. DB Monitor Expert will display editor's popup menu.
3. Choose Select All command then Copy command from the menu. DB Monitor Expert will copy selected source code to the system clipboard.
4. Paste clipboard contents into [DBA Notepad](#) or your favorite SQL Editor.



Brief description of Tab Pages available on the Object Properties pane

- **General** - general information about selected object or object type.
- **Definition** - object definition presented in a tabular format.
- **Code** - reverse-engineering script for the selected object.
- **Interface** - list of functions and procedures in the selected package. This tab is available only connected to Oracle8
- **Navigator** – table/view definition presented in easy-to-view tree-view format. This definition also includes available columns, indexes, constraints, triggers as well as table and index storage options and partitions.
- **Data** - data retrieved from selected table, view or synonym. Note that DB Monitor Expert retrieves as many rows as needed to fill visible tab area. Additional rows are "retrieved as needed" when the user scrolls down to view rows that have not been viewed yet.
- **Security** - grants made on the selected object.
- **Dependencies** - object dependencies for the selected object.
- **Indexes** - list of indexes and their definitions in a tabular format for the selected table.
- **Constraints** - list of constraints and their definitions in a tabular format for the selected table.
- **Triggers** - list of triggers and their definitions in a tabular format for the selected table.

View Database Structure Mode

In this view, Object Navigator displays complete information about logical and physical structure of your database. When viewing Instance parameters, Object Navigator creates General tab page differently. The page is divided into two parts:

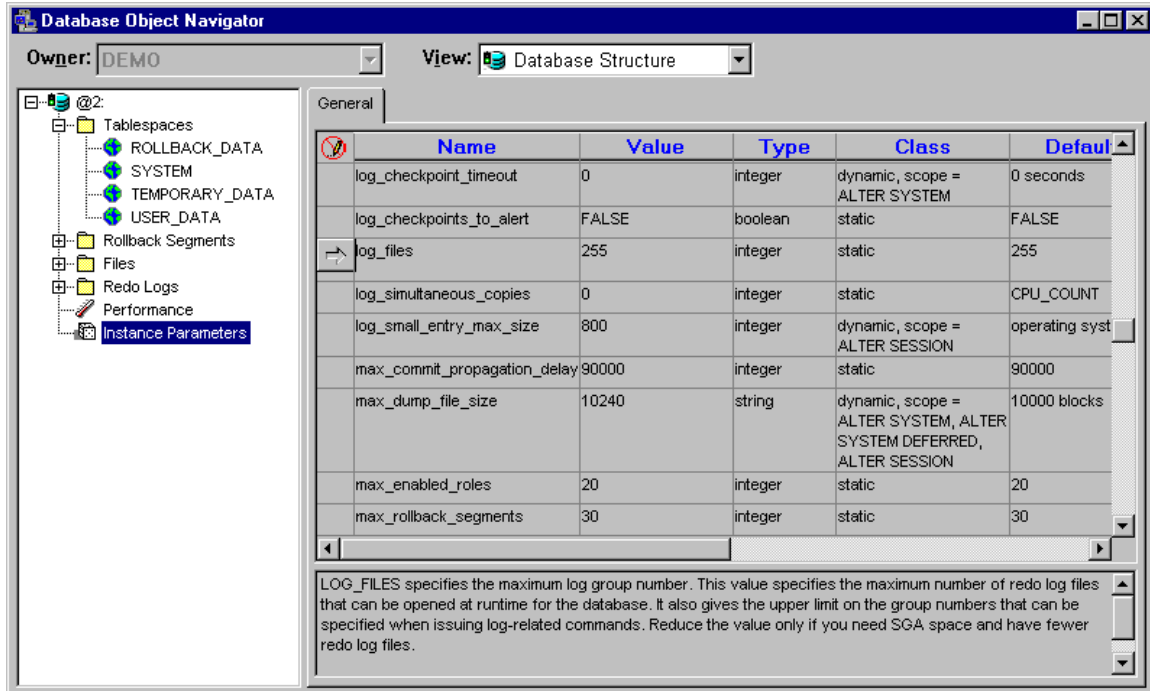
- The "Parameters" - upper pane
- The "Parameter Description" - lower pane

You can change the size of the panes. When you make any pane larger, the adjacent pane becomes smaller. To resize panes:

1. Move the mouse pointer in between of the panes until you see a black line with two arrows.





2. Drag the edge of the pane until the pane is the size you want.



Click on the required parameter to view its description on the lower pane.

DBMS Output

DBMS Output window performs periodical polling of the Oracle DBMS Output buffer. To change polling rate click properties button  on the program toolbar. The default rate is 5 seconds. DBMS Output window is capable to buffer and display up to 999 lines of DBMS Output. If new lines come from the DBMS Output buffer and there is already 999 line in the window then DB Monitor Expert discards as many existing lines as needed for adding new lines. To output some text into DBMS Output buffer you can use Oracle DBMS_OUTPUT package. The DBMS Output window, when used with the [DBA Notepad](#) is an effective combination for developing and debugging stored procedure code.

 **Note:** DBMS Output window must be opened before you send something to the DBMS_OUTPUT buffer. It is a good idea to open this window before you start debugging your procedure code. You should execute your procedure in the same session in order to see the results.. You can use [DBA Notepad](#) for this purpose.

For more details on DBMS_OUPUT package and features see your Oracle documentation.

Schema Reverse-engineering

To reverse-engineer SQL of all schema objects for a particular schema:

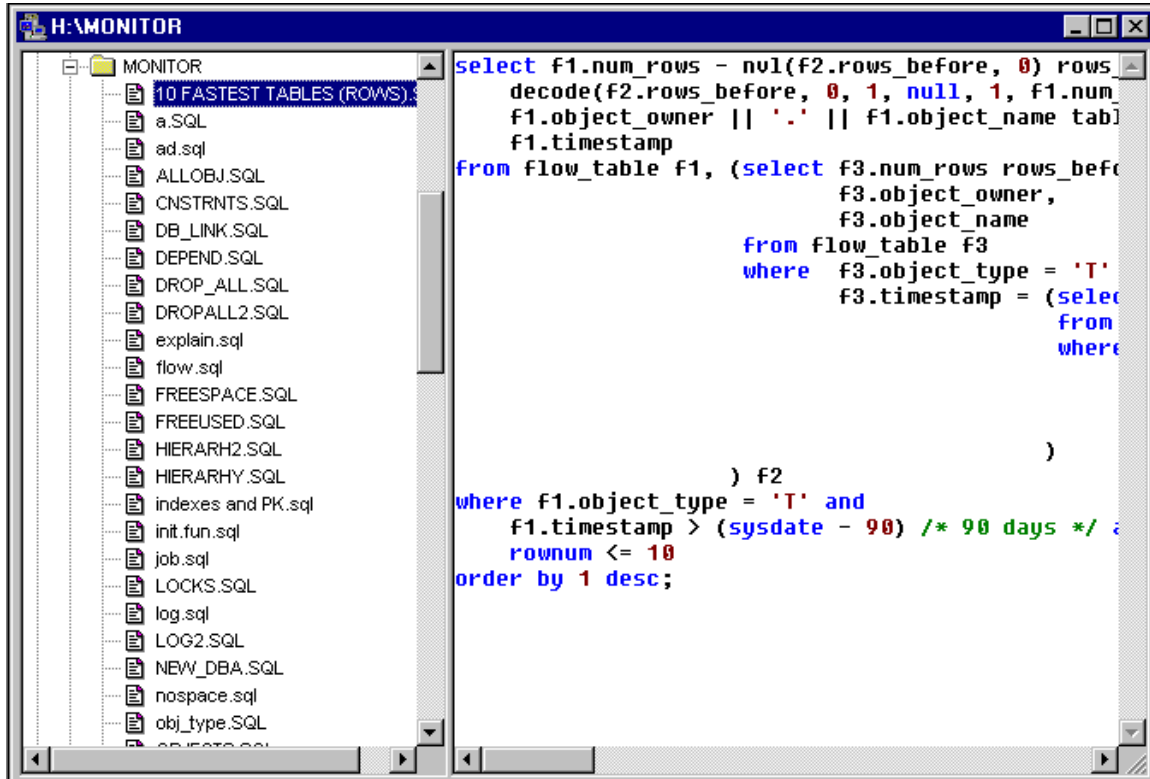
1. Select Tools menu then Schema Extract.
2. Highlight the required schema
3. Enter name of the output file
4. Click OK to start the extraction process.

SQL Navigator


SQL Navigator is a handy tool that you use to browse SQL files on your computer.

The SQL Navigator window is divided into two parts:

- The "File Explorer" - left pane
- The "File Viewer" - right pane



You can change the size of the panes. When you make any pane larger, the adjacent pane becomes smaller. To resize panes:

1. Move the mouse pointer in between of the panes until you see a black line with two arrows.

2. Drag the edge of the pane until the pane is the size you want.



Tips:

- To open a file for editing double-click file name on the left pane.
- Use right-click the File Explorer pane to display context menu. That menu will provide you with additional navigation and file search commands.
- Use right-click the File Viewer pane to display context menu. That menu will provide you with additional commands for text searching and editing.

PL/SQL Debugger

PL/SQL Debugger is an utility that can help you locate the source of logic errors (bugs) in your executable back-end code. It can be used with:

- Oracle server version 7.33 and later
- PL/SQL version 2.3 and later
- Package, Procedure, Function, and Trigger objects

PL/SQL Debugger allows you to debug your back-end code by transferring control of the execution to you. You can stop the execution at any point, execute the code one line at a time, watch how variable values change after a particular statement is executed, find out the order in which statements are called, and see what happens when you change the value of a particular variable. See [Debugger Features](#) topic for additional on what is supported and what is not in different Oracle versions.

PL/SQL Debugger functions

PL/SQL Debugger provides the following functionality to help you debug your code:

- Suspending Execution -- You can suspend code execution before a specific statement is executed in order to examine the state of the object at the point of execution or step through parts of code. This can be achieved by inserting a breakpoint at that line of code containing the statement.
- Stepping Through Code -- Once the code execution is suspended, you can step through the code line-by-line in order to identify the exact location of the problem. This is done by using Step In, Step Over, Step Out, and Run to Cursor procedures to execute required portions of the code.
- Watching for Values of Variables -- You can observe values of specified variables when an execution is broken. This can be achieved by adding variable names to the Watch tab page. Using this tab page, you can also assign a certain value to a variable in order to observe how it will affect execution results.
- Monitoring Execution Flow -- You can monitor execution flow by examining the order of executed statements displayed in the Call Stack tab page.

Using the Debugger

Before starting a debugging session, you should [compile your objects with debugging information](#).

To use the debugger:

1. Open [DBA Notepad](#), and enter the appropriate SQL code invoking the procedure or function that you want to debug.

Example:

```
BEGIN
  SCOTT.RAISE_SALARY(10);
END;
```

2. Select **File/Debug/Start** command from the menu (or press CTRL-F5 keys). The debugger will start separate debugging session and start executing the script. After the debugging session is started, the script execution will pause before the first statement. From this point you can choose the appropriate debugger command from the **File/Debug** menu to continue the script execution. You can also use this pause to setup breakpoints. When the script stops at a breakpoint, you can examine the values of variables and see the call stack. The debugger also lets you change script variables.

Breakpoints

Breakpoint is a line of PL/SQL code in a script at which debugger automatically suspends execution.

To make Debugger pause execution of your code, you can set a breakpoint:

- In the DBA Notepad text pane, double-click on the required line of code that isn't already a breakpoint *or*
- Move the caret to the required line then either press CTRL B key or select File/Debug/Toggle Breakpoint command from the menu.

To remove a breakpoint:

- Double-click the line of code on which the breakpoint has been set
or
- Move the caret to the breakpoint line then either press CTRL B key or select Toggle Breakpoint command from the Debug menu.

A breakpoint can be only set on a line that contains an executable statement

Debugger Views

The debugger has two panes, which you can resize at any time.

Top pane shows the full text of a currently executed script. In that pane, you can go to a specific line in a script, find a string, print the displayed script, and manage breakpoints.

Bottom pane contains 4 tab pages:

Variables Watch - A list of specified variables and their values. The values in the Variables Watch tab page are updated each time the debugger pauses. You can double-click on any variable to manually change the value of a variable whenever script execution is suspended.

Call Stack - The sequence of PL/SQL statement calls leading up to the script that was executing at the time of the breakpoint. The call stack view is automatically refreshed each time the debuggers pauses.

Breakpoints - A list of all breakpoints.

Immediate - A part of the Debugger window in which you can run individual lines of SQL code, usually for the purpose of debugging. The Immediate pane is a kind of scratch pad window in which statements and variables are evaluated immediately. In the Immediate page you can type a statement then select

File/Debug/Execute Immediate command from the menu to execute this statement. You can also use the context popup menu to execute this command as well as copy, paste text and run other common editing commands

Stepping Through Code

Stepping through your script can help you identify where an error is occurring. You can see whether each line of code produces the results you expect. To step through the code after you have it opened in the Debugger, do one of the following:

- To step through each line of script and into the code in a statement called by the active script, select **File/Debug/Step In** command from the menu, or alternatively you can press CTRL F6 keys
- To run the code that precedes the required line of code, and then break so you can step through each line of code, set a breakpoint on that line then select **Start** or **Continue** command from the **File/Debug** menu. The Debugger will suspend execution at the breakpoint line. Use **File/Debug/Step In** command to continue line by line (see previous paragraph for details).
- To run the rest of the current script, select **Continue** command from the **File/Debug** menu. The Debugger will continue executing script until it reaches a breakpoint or an end of script, whatever is first.

The type of stepping you do depends on which portions of code you want to analyze.

Evaluating/Modifying Variables

PL/SQL debugger supports several alternative methods for evaluating values of script variables:

1. You can add variables to the **Variables Watch** tab. The values in the Variables Watch tab page are updated each time the debugger pauses.
or
2. You can move mouse pointer over variable name in the debugged script displayed on the screen and rest it there for a couple of seconds. The debugger will display balloon help over the variable name with the evaluated value.
or
3. You can right-click a variable in the debugged script displayed on the screen and then select **Evaluate/Modify Variable** item in the popup menu.

To modify a variable either of the following methods:

1. Add the variable to the **Variables Watch** tab. Enter new value in the Value field and press the enter key.
or
2. Right-click on a variable in the debugged script displayed on the screen and then select **Evaluate/Modify Variable** item in the popup menu. Enter the new value and click the OK button to close the **Evaluate/Modify Variable** dialog.

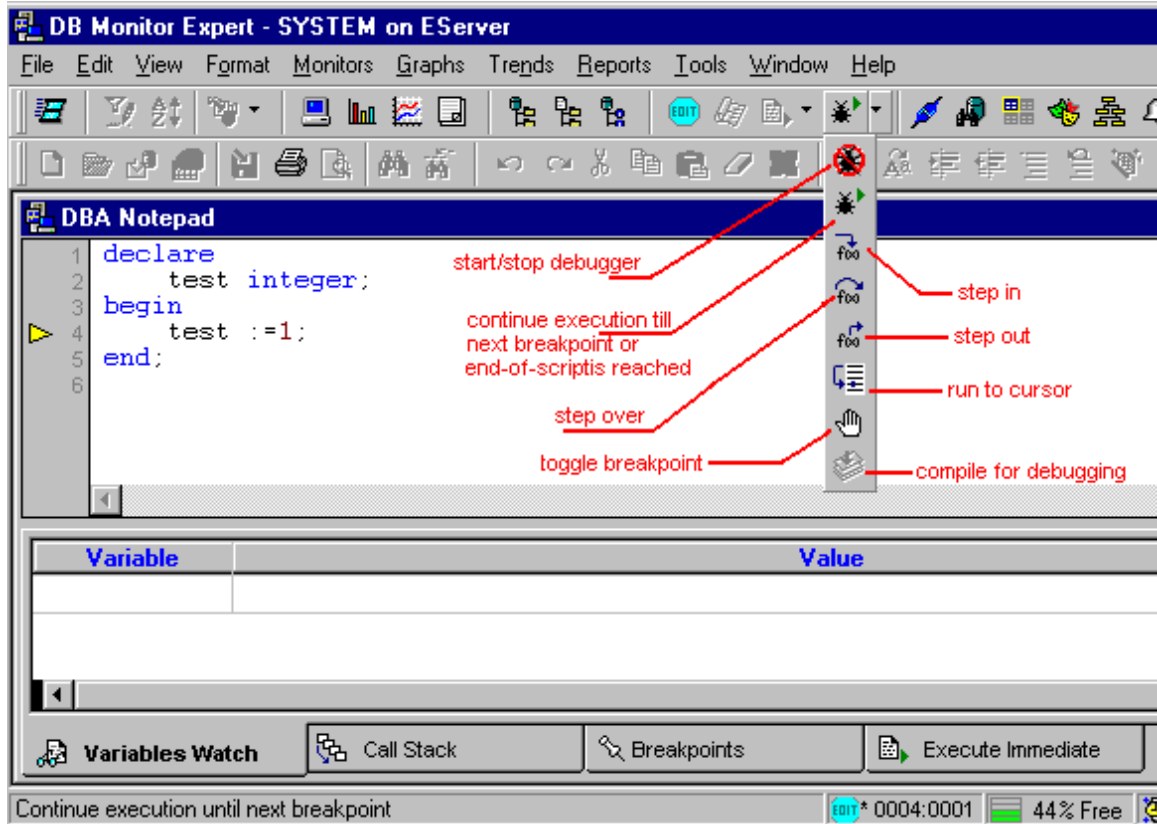
Stopping the Debugger

At any time during a debugging session, you can end the execution by clicking the **File/Debug/Stop** command in the menu button. This will end the PL/SQL Debugger session.

The faster navigation in the debugger you can use hotkeys and also use the debugger's toolbar buttons displayed and commented on the image below

Hot Keys

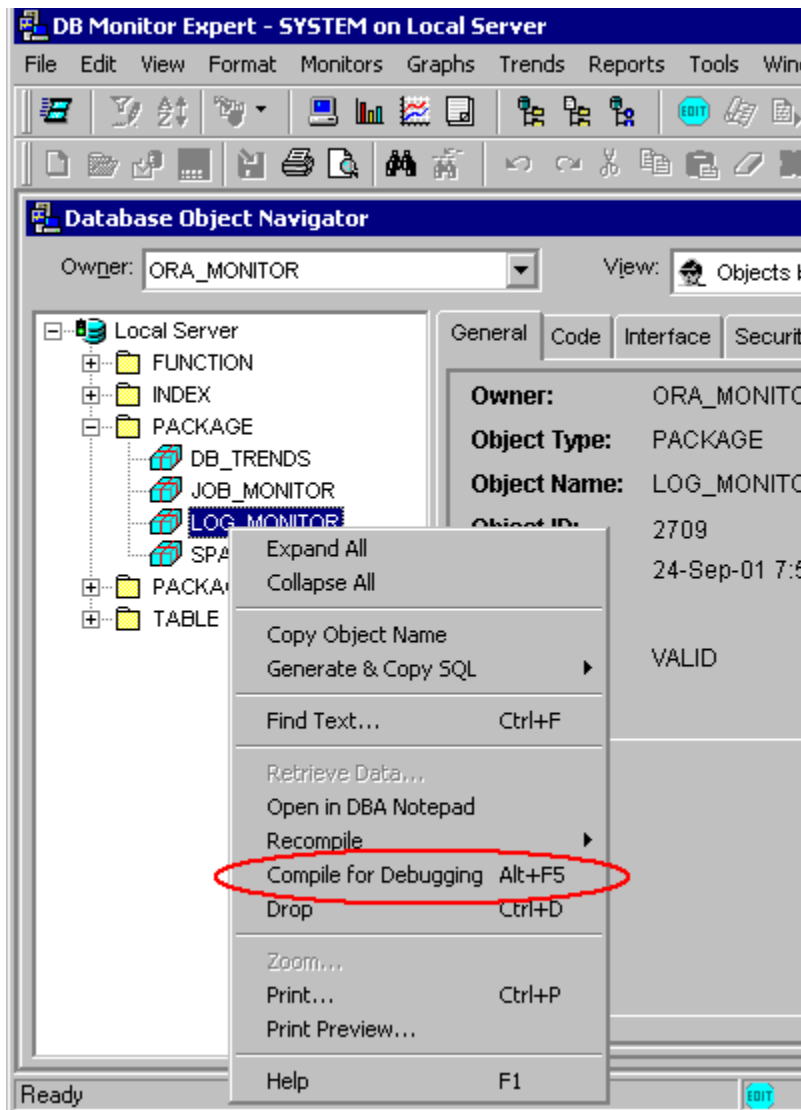
CTRL F5	-- Start/stop debugger
F5	-- Continue script execution until next breakpoint or end-of script is reached
CTRL F6	-- Step into procedure/function call
F6	-- Step over procedure/function call
CTRL B	-- Toggle breakpoint on/off on the line having the caret.



Compiling Objects for Debugging

In order to be able to view/modify variable values during a debugging session, you must compile all required objects with debugging information.

In the [Database Object Navigator](#) select the required object then use **File/Debug/ Compile for Debugging** command from the menu (shortcut ALT+F5). The selected object will be compiled with debugging information included. The Compile for Debugging dialog box opens. This dialog box will allow you to search for other objects that are called within the select object. You can also right-click the required object and select **Compile for Debugging** command from the popup menu.



Version Specific Debugger Features

The DB Tools Debugger uses Oracle Probe API to implement debugging. Therefore, the Debugger is limited by the version of Oracle's Probe API you are using. In addition, some problems in the debugger could be platform and version specific.

Debugger Feature	Oracle Version								
	7.3.2	7.3.3	7.3.4	8.0.3	8.0.4	8.0.5	8i	9i	10g
Debugging supported	✓	✓	✓	✓	✓	✓	✓	✓	✓
View/set values of local variables		✓	✓	✓	✓	✓	✓	✓	✓
View/set values of package global variables						✓	✓	✓	✓
View cursor variables			✓ ¹		✓ ¹	✓	✓	✓	✓
Information on procedures other than current			✓ ²		✓ ²	✓ ²	✓	✓	✓
View/set values of indexed variables									
Stop execution		✓	✓	✓	✓	✓	✓	✓	✓


Notes:

¹ Availability varies depending on platform; a patch may be required

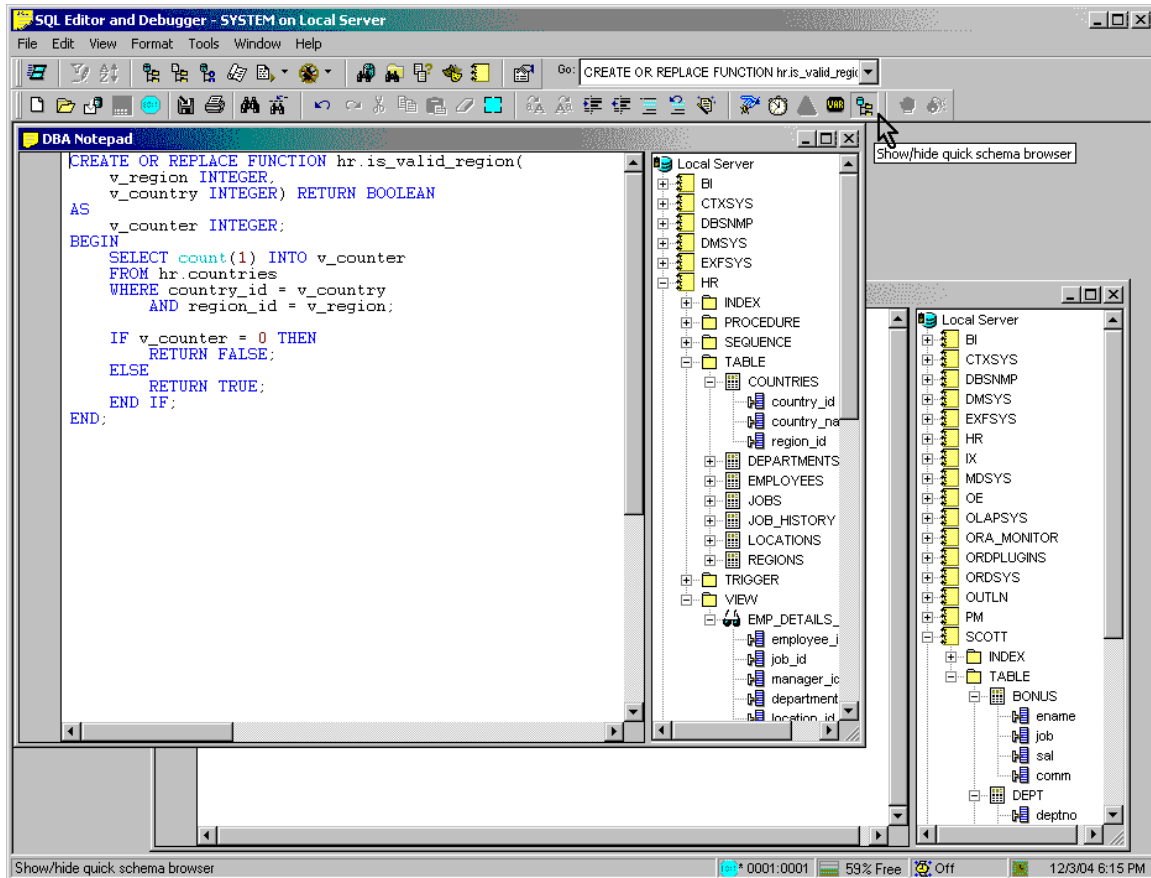
² Requires Oracle initialization parameter `BLANK_TRIMMING=TRUE`.

Using drag-and drop and quick schema browser

You can use the quick schema browser available in each DBA Notepad window to enter schema, object, and column names using standard drag-and-drop feature. Using this feature you can avoid many typo errors.


1. To show or hide the schema browser click **View/Show Schema Browser** menu or click  button on the application toolbar.
2. Navigate to the object such as a table whose columns you want to enter to your script.
3. Drag-and-drop this object to the place in the editor where you want to enter that name.

Below is a sample screenshot demonstrating how to use this feature.



Comparing contents of SQL, TXT and other flat files

To compare 2 text files in SQL Editor:

- 1 Click **Tools/Compare Files** menu or click  button displayed on the SQL Editor toolbar. The **Select Files** dialog will appear.
- 2 Type in or use Browse buttons to select 2 files and then click the **Compare Files** button at the bottom of the dialog window.
- 3 DB Tools will compare selected files and display **Comparison Results** window. Found differences will be highlighted in blue. Total number of found differences will be displayed under each file. If you compare large files use the scroll bar to scroll the **Comparison Results** window.

Customizing DBA Notepad

To change font used to display text in the editor, select **Tools/Options** command. An [options dialog](#) will appear. Click the **Editor** tab and make necessary changes.

Table Relationships

The Table Relationships screen can be used for many purposes. For example it can be used for any of the following:

- Explore design of database tables and views
- Explore and print referential integrity of the database schemas and objects.
- Explore table indexes and their relation to referential constraints.
- Graphically build queries for database tables and view.
- Export data from tables and view to operation system files in a variety of formats.
- Import data from operation system files back to database tables.
- Launch table/view design assistants directly from the Table Relationship workspace.

Below is sample screenshot of the Table Relationships screen with additional comments.

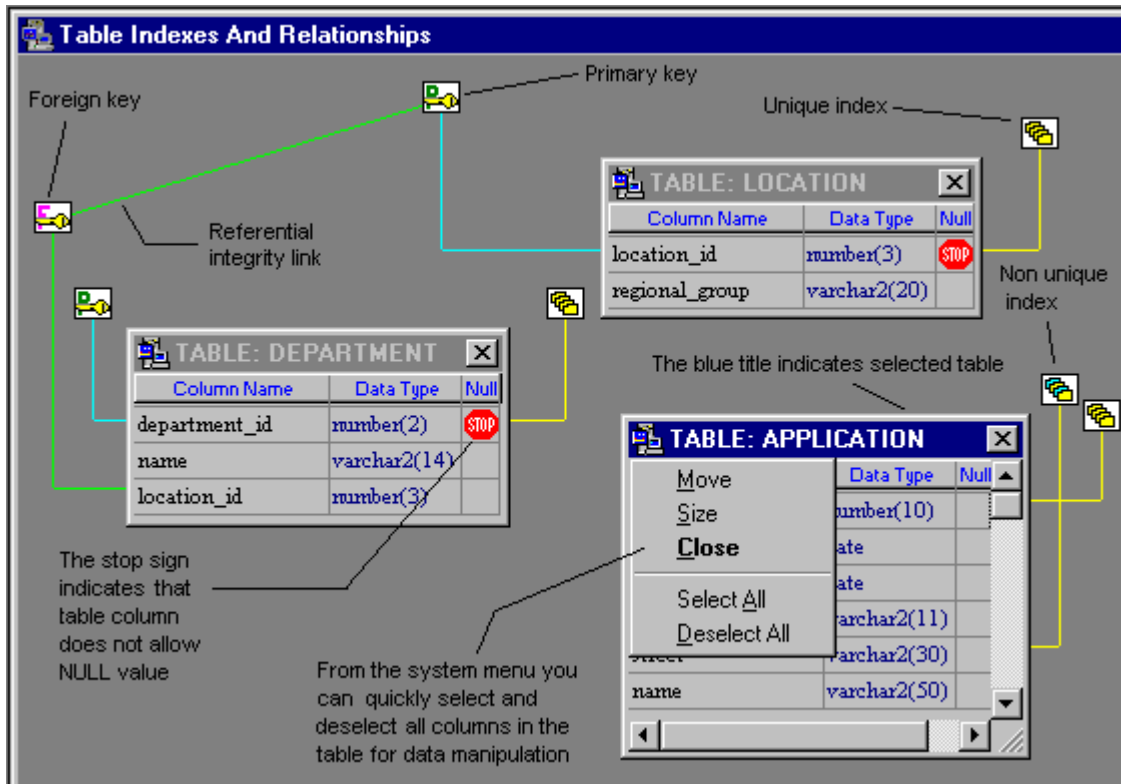


Table Size Estimator

Overview

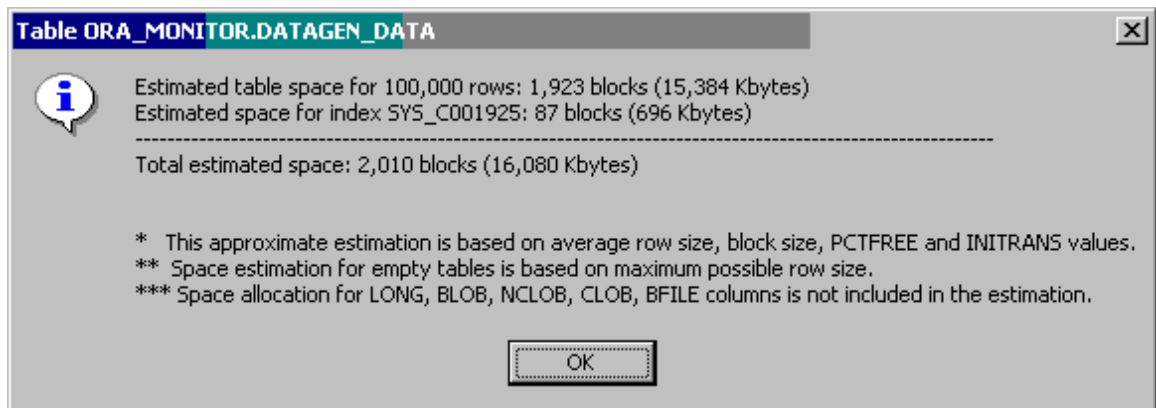
Table Size Estimator is a utility that estimates the space requirements for a table and its indexes based on the projected row count. Table Size Estimator gathers current statistic for the table and its indexes and uses it for approximation of the new total space based on results of the current table data analysis and storage parameters. If the table has been previously analyzed, DB Monitor Expert uses statistics from the Oracle

data dictionary tables. Otherwise, it performs alternative analyzes of existing data from this table. You can use the [Table Analyzer](#) and the [Index Analyzer](#) to analyze tables and their indexes.

Warning: Before you use Table Size Estimator to project new table size, make sure that the table has enough typical data that DB Monitor Expert can use for the analysis.

Space Estimation

5. Select **Tools/Table Size Estimator** command (shortcut: ALT T, E). The **Select Table** dialog will appear.
6. Select schema from the drop-down **Owner** list. DB Monitor Expert will populate list of tables for this schema.
7. Select table name from the list.
8. Enter projected row count in the **Number of Rows to Estimate** field.
9. Click the **OK** button. DB Monitor Expert will perform table analysis and space estimation. After that, a message box will appear where you will see space estimation results for the table data and each index separately as well as the total amount of estimated space.
10. Click the **OK** button to close this result message. You can run the estimate again for a different number of rows or estimate some space requirements for another table.





Database Object Finder

To help you navigate large databases with many objects, DB Monitor Expert includes a Database Object Finder that can quickly search for the database objects that contain a search string in their name.

To Find a Database Object

1. Select **Object/Find Database Object** command from the program menu. The [Select Schema dialog](#) will open.
2. In the **Find What** box, enter the text you want to search for.
3. Select schemas in which search has to be performed.
4. Choose appropriate search options.
5. Click the **OK** button to initiate the search.

 **Note:** To cancel the search operation while it is in progress, click the Cancel  button on the program [toolbar](#).

Browse Search Results

The search results window is divided into two parts:

- "Contents" - left pane
- "Details" - right pane


You can change the size of the panes. When you make any pane larger, the adjacent pane becomes smaller. To resize panes:


1. Move the mouse pointer in between of the panes until you see a black line with two arrows.



2. Drag the edge of the pane until the pane is the size you want.

Use "Contents" pane to navigate search results.

 **Note:** DB Monitor Expert automatically applies filter for selected contents item, so that you see only appropriate items on the "Detail" pane. To see all items, select very first item on the left pane. That item represents all objects in your database.

 **Note:** You can change default filter and sort order for displaying search results. See [Working with data and data export](#) topic for details.

SQL Profiler

Overview

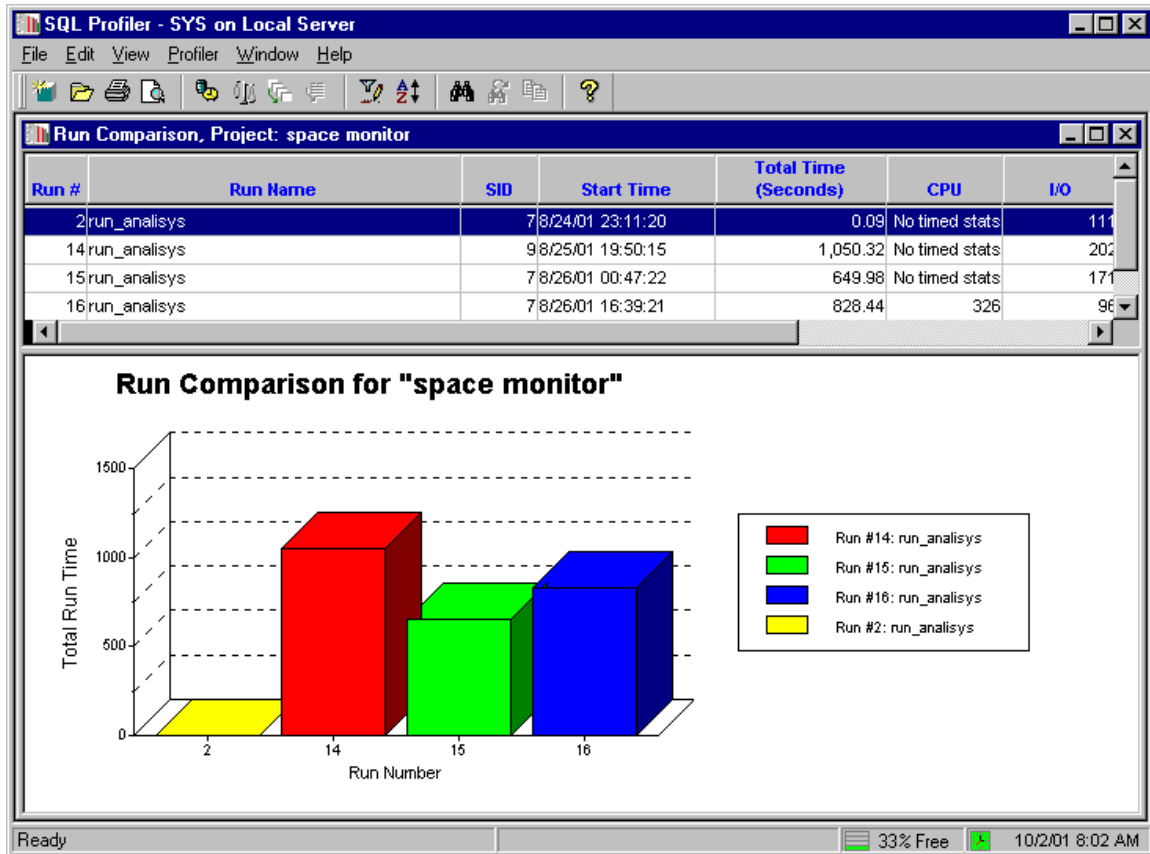
DB Tools PL/SQL Profiler provides graphical interface to the Oracle's Probe Profiler API, which is available in Oracle8i and later. PL/SQL Profiler provides a way to profile existing PL/SQL applications and to identify performance bottlenecks. You can use the collected profiler (performance) data for performance improvement or for determining code coverage for PL/SQL applications. Application developers can use code coverage data to focus their incremental testing efforts.

Improving application performance is an iterative process. Each iteration involves the following steps:

1. Running the application functions with one or more benchmark tests with profiler data collection enabled.
2. Analyzing the profiler data and identifying performance problems.
3. Fixing the problems.

The PL/SQL Profiler supports this process using the concept of a "run". A run involves running the application through benchmark tests with profiler data collection enabled. The profiler writes the collected data for the run into database tables

Because tuning the application is an iterative and a time consuming process, you may need to make incremental changes in your application code and re-run the tests, which can potentially take a long time. To simplify the tuning process DB Tools PL/SQL Profiler provides support for profiler projects. Profiler projects allow grouping of related runs and persistently storing them in the database along with development and tuning comments and PL/SQL code used in each run to invoke the main application code. The projects also store additional performance statistics not available in the Probe API.



PL/SQL Profiler includes several reports that compare performance of individual executed lines of code, performance of callable subprograms and entire runs.

See also:

- [Setting up the PL/SQL Profiler](#)
- [Working with PL/SQL Profiler projects](#)
- [Profiling applications and analyzing results](#)


Setting up PL/SQL Profiler

Requirements

- Oracle 8i and later
- DBMS_PROFILER package and profiling tables

Oracle DBMS_PROFILER package is not installed in the database by default. We recommend that you install this package using PROFLOAD.SQL script that can be found in the [Oracle HOME]\RDBMS\ADMIN directory. This script internally uses DBMSPBP.SQL, PROFTAB.SQL, and PRVTPDB.PLB files that must also exist in the same directory. If you don't these files you can download them from the Oracle TechNet web site <http://technet.oracle.com>.

The PROFLOAD.SQL script installs version of the profiler package and tables that can allow profiling across all users of a system, for example, to profile all users of an application package, independent of who is using it.

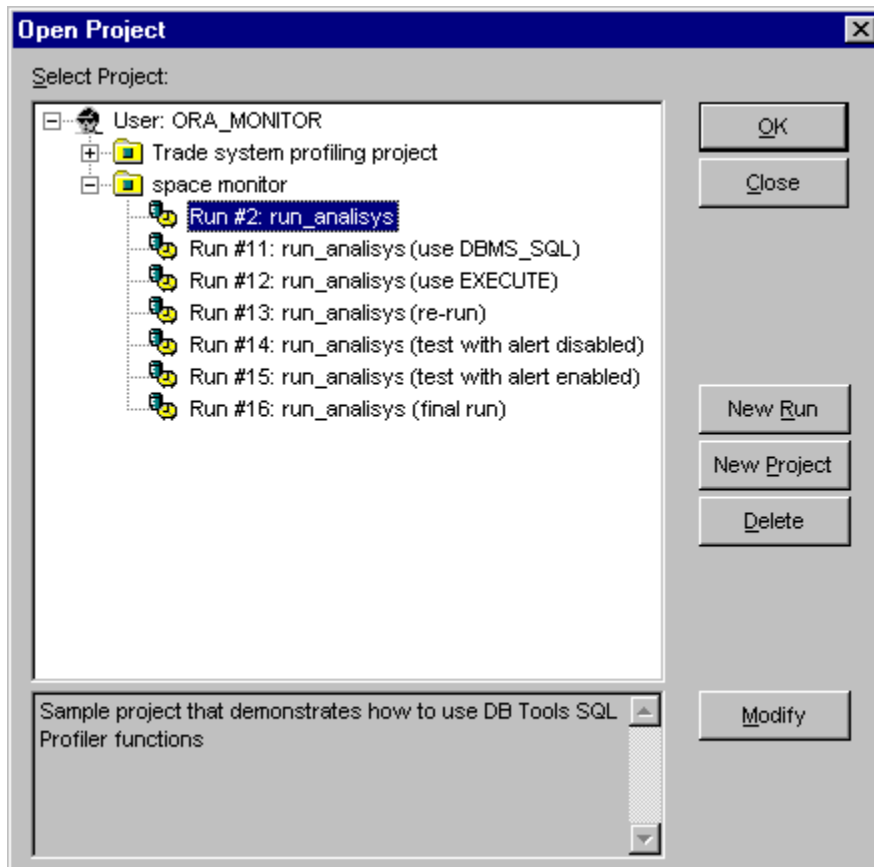
 **Important Note:** Make sure to login as SYS user in order to install DBMS_PROFILER package.

First time you run PL/SQL Profiler it will also create PROFILER_DATA table in the ORA_MONITOR schema. This table is used for storing project and profiling session information and also storing additional profiler statistics not available in the Probe API.

Working with PL/SQL Profiler Projects

In DB Tools PL/SQL Profiler you are always working in the context of a project. Projects allow grouping of related runs and persistently storing them in the database along with development and tuning comments and PL/SQL code used in each run to invoke the main application code. When you create a project, you must name it and optionally you can specify optional project description and project goals. Project information is automatically saved in the database. You can add new *runs* to the project at any time, however the project is saved only after at least one successful run is added. You can create multiple projects for different applications or tuning goals.

Projects are stored in the database you work on. User who created the project is the owner of that project. When you start PL/SQL Profiler the Open Project dialog appears.



The dialog shows only projects that belong to the current user and stored in the connected database. Using this dialog you can select an existing project or create a new one.

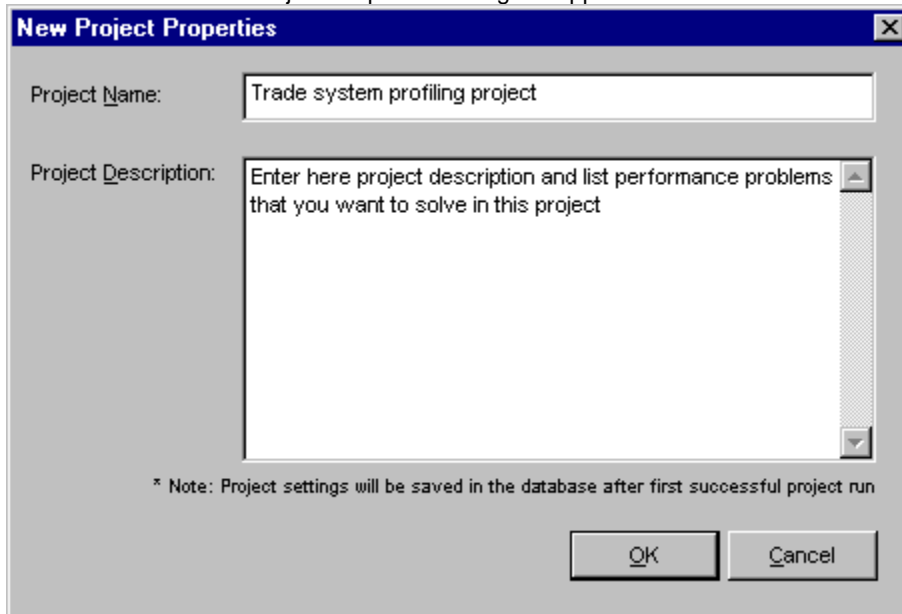
To open a project

1. Use **File/Open Project** menu to open the Open Project dialog

2. Locate the required project in the project tree-view and click Open Project button. The [Run Comparison](#) screen will appear. From this screen you can start new runs and also launch other profile reports.


To start a new project

There exist two alternative ways to create a new project. You can either use **File/New Project** menu or create a new project from the Open Project dialog by clicking on the New Project button. In both cases the New Project Properties dialog will appear.



To complete the dialog:

1. Enter project name, this name should be unique and descriptive and can include any characters including spaces.
2. Enter optional project comments
3. Click the OK button.

 **Important Note:** The new project definition is saved in the database only after the first project run. If no run was completed, the project definition will be lost when you exit PL/SQL Profiler.

To modify a project

1. Use **File/Open Project** menu to open the Open Project dialog
2. Locate the required project in the project tree-view and click Modify button. The project properties dialog will appear.
3. Correct the required properties and then click the OK button, or click Cancel to cancel the operation

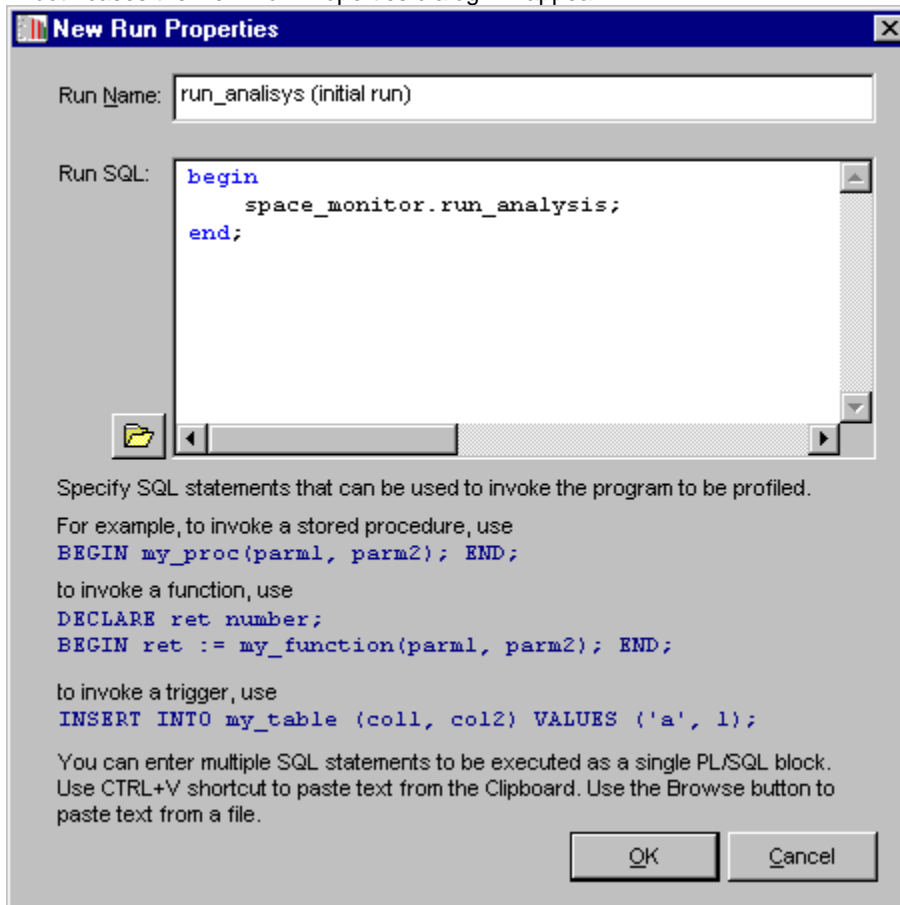
To delete a project

1. Use **File/Open Project** menu to open the Open Project dialog
2. Locate the required project in the project tree-view and click Delete button
3. The confirmation message will appear. To delete the project click OK, or click Cancel to cancel the operation

To start a new run

There exist two alternative ways to start a new run. You can either use **Profiler/New Run** menu or start a new run from the Open Project dialog by clicking on the New Run button.

In both cases the New Run Properties dialog will appear.



To complete the dialog:

1. Enter descriptive run name, this name descriptive enough to allow you later easily identify this run in the project tree-view. This name can include any characters including spaces.
2. Enter SQL code than will invoke the program that you want to profile.
3. Click the OK button.

To delete a run from project


1. Use **File/Open Project** menu to open the Open Project dialog
2. Locate the required project in the project tree-view, expand it and then select the run that wish to delete.
3. Click Delete button. The confirmation message will appear. To delete the run click OK, or click Cancel to cancel the operation


Profiling Applications and Analyzing Results

Using PL/SQL Profiler provides you can generate different kinds of profiling information:

- Profiling information for runs, such total run duration, CPU used by the session, amount of memory used, number of disk I/O operations, and so on.
- Profiling information for all named library units that are executed in a session, such as number of times a unit has been executed, the total amount of time that has been spent executing that unit, and the minimum, maximum and average times that have been spent executing a particular object.
- Profiling information for all code lines that are executed in a session, such as the total number of times each line has been executed, the total amount of time that has been spent executing that line, and the minimum and maximum times that have been spent on a particular execution of that line.

The collected statistics are analyzed and presented in easy-to-read profiler reports. Every profiler report contains two views to the collected data: the data-grid view and the top statistics graph.

 **Important Note:** Because of the security considerations, the profiler only gathers data for units for which a user has CREATE privilege; you cannot profile programs for which EXECUTE ONLY access has been granted. In general, if a user can debug a unit, the same user can profile it. However, a unit can be profiled whether or not it has been compiled for debugging.

 **Tip:** Oracle advises that modules that are being profiled should be compiled with debugging flag turned on, since this provides additional information about the unit in the database. You can use the DB Monitor Expert module to compile objects for debugging. See [Compile For Debugging](#) topic for "How To" information.

Analyzing results

PL/SQL Profile features 3 different reports that can be used to analyze and compare profiler results. These reports include:

- Run Comparison
- Object Execution Comparison
- Line Execution Comparison

These reports provide 3 different view levels. By double clicking on the required report row or bar in the report chart you can drilldown to the detail report for the activated item. From the Run Comparison report you can drill-down to the Object Execution Report and from there you can drill-down to the Line Comparison Report.

To navigate between these reports you can use double-clicks as mentioned above. You can also use the Profile menu to run a detail report for the selected item.

 **Tips:**

- The top portion of each report shows all collected profiler data for the selected run/item. The data is displayed in a tabular format. You can sort and filter data in this portion of the report using **View/Sort** and **View/Filter** menu commands.
- The bottom portion of each report shows "top 10" graph that includes only the top worst runs, objects or lines. When a filter is set on the report, the graph is automatically rebuilt to reflect the changes.
- To quickly located a specific text on the report use **Edit/Find Text** and **Edit/Find Next** menu commands
- To print a report, use **File/Print** menu command

Java Editor

Overview

DB Tools Java Editor is a multi-functional multi-document Java development environment with a built-in file explorer.

You can use this tool to edit regular Java files and also to install Java classes in Oracle databases supporting Java objects such as Oracle 8i and later. The editor features Java syntax highlighting and numerous productivity enhancements. It also provides integrated interface for Java class compilation and direct loading of compiled classes and resources into Oracle databases.

Before you compile your Java classes in your Oracle database, make sure it has Oracle Java environment and compiler installed. If you get *ORA-29547 Java system class not available: oracle/aurora/rdbms/Compiler* or a similar error, it means that the Java system classes are not installed. Login as SYS or SYSTEM user and run "CREATE OR REPLACE JAVA SYSTEM" command from DBA Notepad or from SQL*Plus. This command will take awhile to run, but it should fix your problem.

See also:

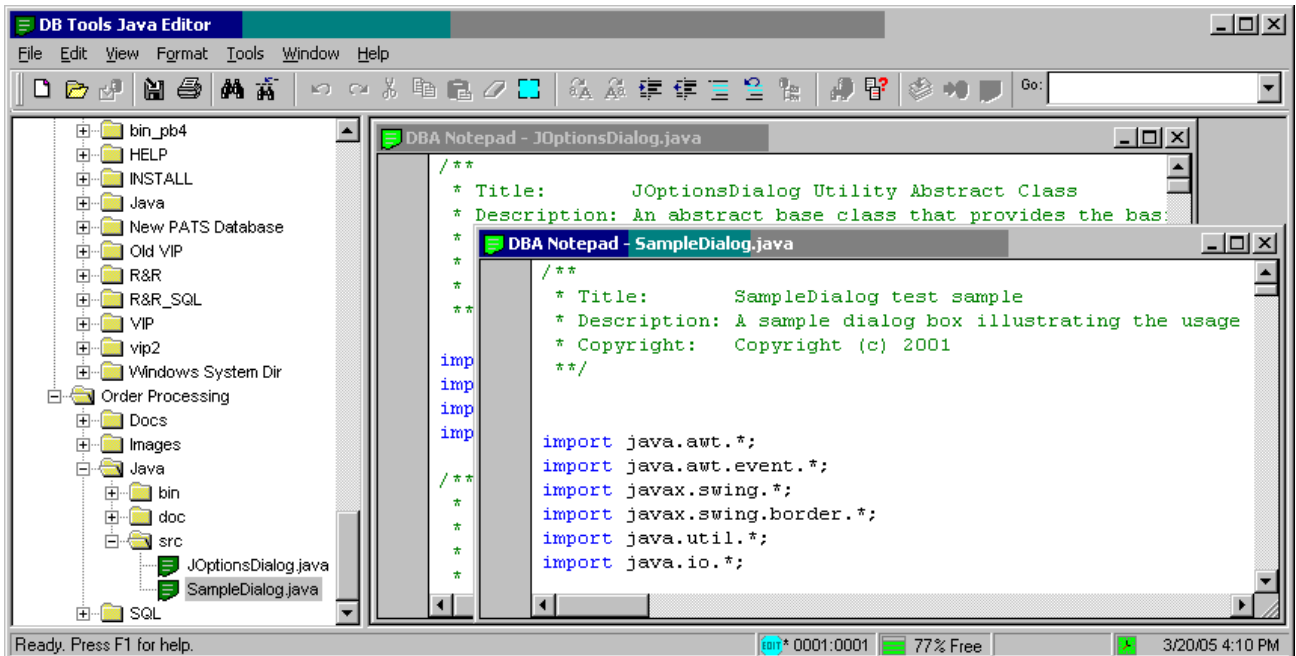
[Java Editor Functions](#)

[Compiling and Loading Java Objects Into Database](#)


Java Editor Interface

The Java Editor window is divided into two parts:

- The "Java/JSP File Explorer" - left pane
- The "Workspace" - right pane



You can change the size of the panes. When you make any pane larger, the adjacent pane becomes smaller. To resize panes:

3. Move the mouse pointer in between of the panes until you see a black line with two arrows.

4. Drag the edge of the pane until the pane is the size you want.

**Tips:**


- To open a file for editing double-click file name on the left pane.
- Use right-click the Java/JSP File Explorer pane to display context menu. That menu will provide you with additional navigation and file search commands.
- Use right-click the Workspace pane to display context menu. That menu will provide you with additional commands for text searching and editing.

Java Editor Functions


Java Editor functions are pretty much the same as the SQL Editor functions. For the command reference see [Editing and Formatting Commands](#) topic for DBA Notepad. For keyboard shortcuts reference see [Hot Keys](#) topic.

Compiling and Loading Java Objects Into Database

Compiling Java class on the Oracle server from a Java a source code

1. Open a file containing required Java sources code classes or enter Java code into the Java Editor
2. Use **File/Compile Java Class** menu or click the  button on the toolbar. The Compile Options dialog will appear.
3. If necessary, correct default compile options and click the OK button to start the compilation.

Load a Java class already compiled on the client

1. Click **File/Load Class File into Database** menu or click  button on the toolbar. The standard Open File dialog will appear.
4. Locate the required Java class file or a file containing Java classes such as JAR or ZIP file and press the OK button. The Compile Options dialog will appear.
5. If necessary, correct default compile options and click the OK button to start the compilation.

File Loader

Overview

File Loader is a simply utility that allows loading files into existing and new database tables. This utility complements several other file loading methods available in SQL Editor and DB Monitor Expert components.

While SQL Editor's and DB Monitor Expert's file import facilities can be used to import file data into existing tables only the File Loader utility supports both loading files into new and existing tables. New tables can be created dynamically during file load process.

The following file types are currently supported:

- XLS - Microsoft Excel files.
- TXT – Tab separated text files.
- CSV – Coma-separated text files.

For details on how to use File Loader utility see [Loading Files into Database](#) topic.

Loading Files Into Database

Loading files into existing tables

To load files into existing database tables use one of the following methods:

Using SQL Editor

1. Execute **SELECT * FROM <table>** command to open a new datasheet. If you do not want to retrieve existing data you can execute **SELECT * FROM <table> WHERE 0 = 1** command to create an empty datasheet.



Note: If the import file does not contain all table columns select only columns contained in the file. Make sure you include all not nullable columns; otherwise an error will occur when you attempt to save the imported data.

2. Click **File/Import** menu to select an import file. Choose the required file name and type and then click **OK**.
3. After the import is complete verify imported data and then click **File/Save** menu to send the imported data into database.



Note: This method is the fastest method in DB Tools to load small and medium size files. Do not use this method for large files as the file is loaded entirely into the computer memory first and only then it is sent to the database.

Using DBA Notepad in DB Monitor Expert

Follow the same steps as described above for the SQL Editor.

Using Table Painter tool in DB Monitor Expert

Using Table Painter tool you can load data files directly into database tables without using datasheets and **SELECT** commands. To load a file:

1. Open Table Painter tool.
2. Select target tables. If you select multiple tables, after these tables are painted click on the table into which you want to import your first file.
3. Click **File/Import** menu to select a file. Choose the required file name and type and then click **OK**.
4. Repeat steps 3 as needed for other tables and files.



Note: This method has no limitation on the file size

Using File Loader utility

1. Start **File Loader** utility.
2. Type in or use the **Browse** button to enter the source data file.
3. Click **Load into existing table** option.
4. Select target **schema** and **table**.
5. Click **Load File** button to start the processing.



Note: This method has no limitation on the file size. This method also allows users to interrupt file load process at any time. Click the **Cancel** button to interrupt the load process at any time.



Note: In order to load Excel files you must have Microsoft Excel version 97 or better installed on your computer.

Loading files into new tables:

1. Start **File Loader** utility.
2. Type in or use the **Browse** button to enter the source data file.
3. Click **Load into new table** option. Enter target table name. Table name can optionally include schema name in standard Oracle dot notation format. If you do not specify schema name new table will be created in the schema of the user currently logged on.
4. Click **Load File** button.
5. The File Loader utility will analyze your source file and prompt you to verify and correct target table definition. After correcting table definition click the OK button to create the new table and start the file load process.



Note: This method has no limitation on the file size. This method also allows users to interrupt file load process at any time. Click the **Cancel** button to interrupt the load process at any time.



Note: In order to load Excel files you must have Microsoft Excel version 97 or better installed on your computer.

LOB Viewer and Loader

Overview

LOB Viewer and Loader can be used for browsing and viewing various documents saved in database LOB (Large Object) columns as well as for loading new documents into the database. LOB Viewer is especially helpful to database developers because it enables them to view data stored in LOB columns which cannot be done using regular database development tools such as SQL*Plus and most other. With LOB Viewer and Loader you can now view, delete, update and insert almost any document or file directly without launching the document's native application originally used to save the data in the database LOB!

LOB Viewer and Loader supports all types of LOB columns currently supported by Oracle including:


- **LONG**
- **BLOB**
- **CLOB**
- **NCLOB**
- **LONG RAW**
- **BFILE**

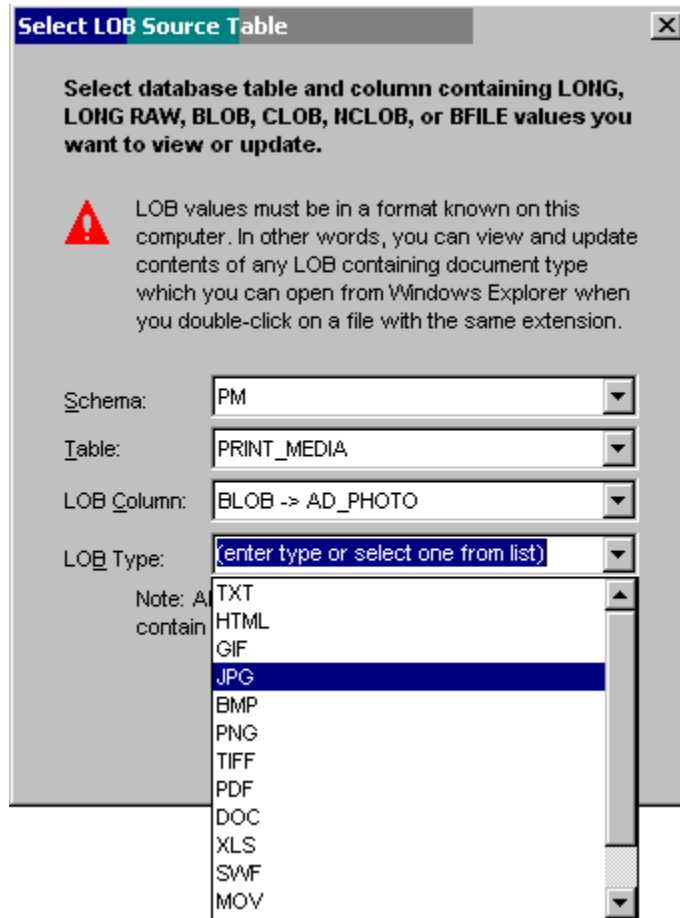
As long as you have any appropriate file viewing application installed on your client computer you can view and edit values saved in such columns. For example you can view and edit text documents, HTML files, a wide range of graphic files, if you simple have a web browser installed on your computer. You can even view Microsoft Word and Excel documents without launching Word or Excel if you have Microsoft Office installed on your computer.

LOB Viewer and Loader is a generic document management application. It is **not** programmed for specific documents of file types. You select at the time of database browsing which document types you want to view or edit and enter your choice into the list of options if the type is not listed or simply select a pre-configured type from the list. LOB Viewer and Loader internally uses COM automation available on your Windows system to start the document handling application (such as web browser or Microsoft Word) inside LOB Viewer window and then controls the started application.

Open LOB Table for Browsing

LOB Viewer and Loader provides very straight-forward interface for the table browsing. To open a table for LOB browsing or editing do the following:

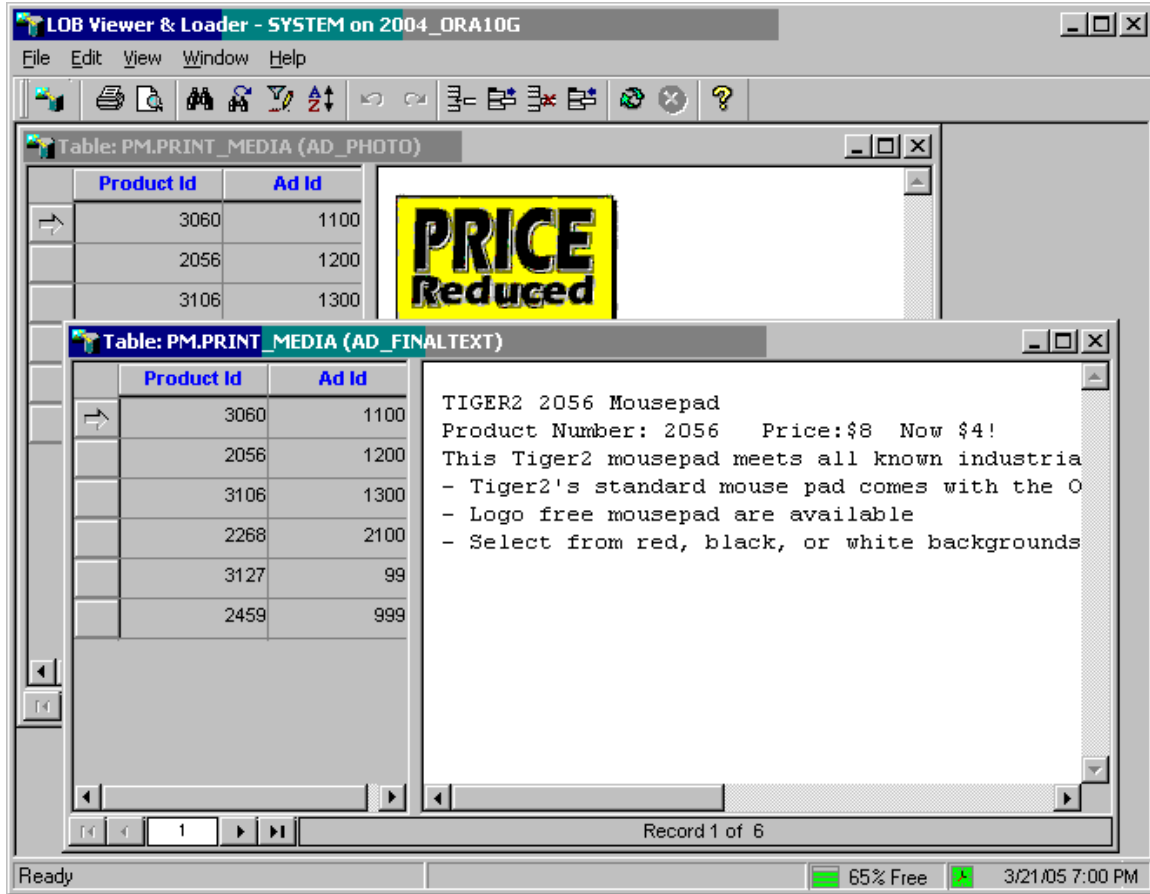
1. Select **File/New Viewer** command (Shortcut CTRL O) from the LOB Viewer and Loader menu.
This will open the **Select LOB Source Table** dialog. Alternatively you can click the Open  button on the application toolbar.
2. Using the **Schema** drop-down box select table schema containing the required table with a LOB column. The **Table** drop-down gets populated with the names of tables in the selected schema.
3. Using the **Table** drop-down select the required table. The **LOB Column** drop-down gets populated with the names of table columns having one of the supported LOB types.
4. Using the **LOB Column** select the required column. Please note that available LOB columns are displayed in <LOB Type> -> <Column Name> format that makes it easier for you to decide which types of documents are stored in these columns.
5. If the selected document type is available in the **LOB Type** drop-down list you can select it from the list otherwise simply type the document type as if was a file extension. For example, if the document type is XML document you would type XML in the **LOB Type edit field**.



6. Click the OK button to open the table datasheet and close the **Select LOB Source Table** dialog.
7. Browse or modify data using available table datasheet functions. See [Browsing and Modifying LOB Tables](#) topic for additional information

Browsing and Modifying LOB Tables

After you get your LOB table [opened for browsing](#) DB Tools will display LOB table datasheet with the table data. The following is a sample screenshot containing 2 datasheets opened simultaneously for the same table. Because LOB Viewer and Loader is an MDI (multi-document interface) application you can actually open as many datasheets as needed. Multiple datasheets can be opened for the same or different LOB tables.



The datasheet window is divided into two parts:

- The "Data Table" - left pane.
- The "Viewer" - right pane

You can change the size of the panes. When you make any pane larger, the adjacent pane becomes smaller. To resize panes:

1. Move the mouse pointer in between of the panes until you see a black line with two arrows.




2. Drag the edge of the pane until the pane is the size you want.


All non-LOB values are displayed on the left pane. LOB document from the chosen LOB column is displayed on right pane. Immediately after datasheet opening the LOB from the first row is shown on the screen. To view a different LOB change the current row on the left pane



Tips:

1. The current row on the left pane always has arrow symbol displayed over the row header. 
2. Scrollbar on both panes disappear and appear automatically whenever the displayed LOB doesn't fit in the pane visible area.

Inserting a new record into a LOB table

1. Select **Edit/Add** menu (Shortcut CTL N) or click the Add  button on the application toolbar. The **New LOB Record** dialog will appear.

- In the top portion of the dialog fill in values for all non-LOB columns.


- In the bottom portion of the dialog enter name of the file containing the document you want to save in the new record.



Important: If the LOB column type is non-BFILE the document must be available on your computer or be accessible on a shared network drive. In that case use the **Internal LOB Storage** tab page to specify the document file name. If the LOB column type is BFILE the document must be available on the database server or be stored on a shared network drive that is accessible from the database server computer. In that case use the **Internal LOB Storage** tab page to specify the document file name using Oracle existing DIRECTORY objects. For your convenience DB Tools provides in place New Directory button that can be used to create new DIRECTORY objects, as you need them. You must have sufficient privileges to create both the DIRECTORY object and access the referenced file path in order to insert new BFILE documents.

- Click the **OK** button to close the **New LOB Record** dialog and insert the new record to the database. The inserted record will automatically appear on the datasheet.

Updating an existing record in a LOB table


- In the LOB table datasheet click on the record you want to update.
- Select **Edit/Update** menu (Shortcut CTL U) or click the Update  button on the application toolbar. If the LOB column type is non-BFILE the standard **Browse for Files** dialog will appear, otherwise the **Select BFILE Properties** dialog will appear.
- If you get the **Browse for Files** dialog
 - Use standard navigation techniques to locate and open the required file.


or

If you get the **Select BFILE Properties** dialog:


- 3.1 Select the **DIRECTORY** object referencing to the directory on the server containing the required document file.
 - 3.2 Enter the new document file name into the **File Name** field.
4. Click the OK button to close the dialog and update current record with the chosen document

Deleting an entire record from a LOB table

 **Important:** Deleting an entire record cause all values from all columns in the selected record to disappear from the database.

1. In the LOB table datasheet click on the record you want to delete.
2. Select **Edit/Delete** menu (Shortcut CTL D) or click the Delete  button on the application toolbar.

Deleting a document from a LOB table

 **Important:** Deleting LOB document **does not delete the record containing that document**. It only clears the value of the selected LOB column, which effectively removes the document from your database.

1. In the LOB table datasheet click on the record associated with the document you want to delete from the database.
2. Select **Edit/Clear** menu (Shortcut CTL L).

Searching documents in a LOB table

You cannot search documents using document internal data because LOB columns contain only binary representations of LOB documents. However you can search records associated with your documents using other columns in the LOB table. The LOB Viewer and Loader provides standard set of search functions that can be used for that purpose. For more information see the general [Searching](#) topic.

Sorting and filtering documents in a LOB table

You cannot use standard DB Tools [Search](#) and [Filter](#) options to control which records are displayed on a LOB table datasheet.

Printing documents in a LOB table

1. In the LOB table datasheet click on the record associated with the document you want to print.
2. Choose **File/Print** menu (Shortcut CTL P) to print the active document or choose **File/Print Preview** menu to see the document on the screen how it will look when printed.

DB Tools Upgrade Assistant

When to Use Migration Assistant

The DB Tools Upgrade Assistant helps in converting existing libraries of user-defined objects created in previous versions of the DB Tools or ORACLE Monitor Suite software to the current version. It can be also used to copy objects between computers. For example if your colleague created a report or monitor that you want to share you can use the Migration Assistant to copy objects from your colleague's system.

The DB Tools Upgrade Assistant also supports converting of the old DB Tools Active Desktop configuration.



Warning: Always backup files having LIB extensions prior to migrations and software upgrades.

Using the Migration Assistant:

- 1 Start DB Tools Upgrade Assistant
- 2 Type in or use the Browse button to select the library of user-defined objects. List of available objects appears.
- 3 Highlight objects that you want to convert and copy to the current version.
- 4 Click Proceed.

Object Status

- **New** - objects having this status do not exist in the default library CUSTOM.LIB.
- **Update** - objects having this status do exist in the default library CUSTOM.LIB, but they either later creation date or they have the same creation date but syntactically different.
- **Old** - objects having this status exist in the default library CUSTOM.LIB and they have earlier creation date.
- **<empty>** - matching objects exist in the default library CUSTOM.LIB.

Miscellaneous Information

Technical Support

Your questions, comments, and suggestions are welcome.

For technical support email to support@softtreetech.com or use the on-line support form at <http://www.SoftTreeTech.com/Support.htm>.

Please use the [Technical Support Request Form](#) when contacting us by email or fax.

When reporting problems, please provide as much information as possible about your problem. Be sure to include the following information:

- 1 Is the problem reproducible? If so, how?
- 2 What version of Windows are you running? For example, Windows XP, Vista 64-bit, etc.
- 3 What version of the DB Tools are you running?
- 4 If a dialog box with an error message was displayed, please include the full text of the dialog box, including the text in the title bar.
- 5 If the problem involves an external program, provide as much information as possible on this program?

Also, make sure to include the serial number for your copy of the DB Tools. Use **Help/About** menu to look up the correct numbers. Registered users have priority support.

For registration information, purchasing or other sales information, please contact our sales department: sales@softtreetech.com

For general information, software updates, the latest information on known problems, and answers to frequently asked questions visit the DB Tools home page on the Web: <http://www.SoftTreeTech.com/dbtools/index.htm>.

We're happy to help in any way we can, but if you're having problems please check the troubleshooting section first to see if your question is answered there.

At SoftTree Technologies, we know that our success depends on how successful you are in your business.

Technical Support Request Form

EMAIL TO: support@softtreetech.com

Please include the following contact information:

Name _____
 Company _____
 Address _____
 City, State/Zip _____
 Phone _____ Fax _____
 CompuServe/Email _____
 Best Time to Reach You _____
 Operating Systems Information:
 Which operating system are you using: _____

Description of the problem:

Describe steps have you taken to try to solve it:
