

# Performance Explorer-i for Oracle

## User's Guide

Version 4.0.1



**DBA**  
**INFOPOWER**

© Copyright DBA InfoPower, Inc. 2007. All rights reserved.

This guide contains proprietary information, which is protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of DBA InfoPower, Inc.

### **Warranty**

The information contained in this document is subject to change without notice. DBA InfoPower, Inc. makes no warranty of any kind with respect to this information. DBA INFOPOWER, INC. SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTY OF THE MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. DBA InfoPower, Inc. shall not be liable for any direct, indirect, incidental, consequential, or other damage alleged in connection with the furnishing or use of this information.

### **Trademarks**

Performance Explorer-i for Oracle is a registered trademark of DBA InfoPower, Inc. Other trademarks and registered trademarks used in this guide are property of their respective owners.

DBA InfoPower, Inc.

666 Pomeroy Avenue

Santa Clara, CA 95051

Web site: [www.dbainfopower.com](http://www.dbainfopower.com)

Email: [info@dbainfopower.com](mailto:info@dbainfopower.com)

Updated February 2007

# Contents

<b>Chapter 1</b>	<b>Introduction</b>	<b>3</b>
<b>Chapter 2</b>	<b>Welcome to Performance Explorer-i for Oracle</b>	<b>6</b>
<b>Chapter 3</b>	<b>Getting started with Performance Explorer-i for Oracle</b>	<b>16</b>
<b>Chapter 4</b>	<b>Common Control Elements of Performance Explorer-i for Oracle</b>	<b>19</b>
<b>Chapter 5</b>	<b>Using Performance Explorer-i for Oracle</b>	<b>25</b>
<b>Chapter 6</b>	<b>Troubleshooting Performance Explorer-i for Oracle</b>	<b>61</b>

# 1

## Introduction

This chapter describes how to use this guide. It also explains how to contact DBA InfoPower and get technical support.

In this chapter	Page
Guide Special Conventions	3
Guide in PDF format	4
Additional information?	4, 5

### Special conventions used in this guide

This guide uses the following typographic conventions:

Example	Description
The <b>Chart</b> button	Bold is utilized for the literal names of interface objects used to perform actions (for example, toolbar button names, menu names, icon names, dialog box options, and dialog box titles).
View -> Swap Panels	The arrow symbol indicates menu path. In this example, choose View, and then choose Swap Panels from the View menu
<i>Variable</i>	italic type indicates a placeholder for information that needs to be provided

Example	Description
Press ENTER	Names of keyboard keys appear in capital letters.
Press ALT + F1	A plus sign (+) between key names indicates that keys are to be pressed in combination. For example, ALT+F1 means to hold down the ALT key while pressing the F1 key.
Press the DOWN ARROW key	Direction keys are referred to by the direction of the arrow LEFT, RIGHT, UP or DOWN keys

## Guide in PDF format

This guide is provided in Adobe Acrobat portable document format. The PDF file is included in Performance Explorer-i installation package. To view the PDF file, you need Adobe Acrobat Reader version 6.0 or higher. Adobe Acrobat reader is available on the Adobe Web site at: <http://www.adobe.com/>

## Additional information

### DBA InfoPower product information

You can contact DBA InfoPower for product information in any of the following ways:

Web page: <http://www.dbainfopower.com>

E-mail (sales & marketing) [sales@dbainfopower.com](mailto:sales@dbainfopower.com)

### Contacting customer support

DBA InfoPower support team is dedicated to ensuring successful product installation and use for all DBA InfoPower solutions.

Support Link: [http://www.dbainfopower.com/dbaip\\_support.php](http://www.dbainfopower.com/dbaip_support.php)

E-mail: [support@dbainfopower.com](mailto:support@dbainfopower.com)

### Performance Explorer-i support

Should you encounter any problems with Performance Explorer, follow these steps:

1. Copy contents of the Performance Explorer-i execution console to a diagnostics text file
2. Capture picture of an error to an image file
3. Copy contents of the Performance Explorer-i error dialog to the diagnostics text file
4. E-mail these files to [support@dbainfopower.com](mailto:support@dbainfopower.com) with your request for assistance

## **DBA InfoPower, Inc. enables business continuity**

In today's economy, businesses rely on information contained in their databases to take orders, process requests and provide information to their employees, customers and partners. Accordingly, it is vital for businesses to ensure uninterrupted reliable operation of their databases.

Databases have, however, become increasingly complex and hard to manage. There is an unprecedented amount of data stored by today's enterprises, creating an increasing strain on their databases. Databases are also being accessed more frequently for a growing number of purposes.

The combination of database importance and complexity of managing database infrastructure has created a demand for products that can improve database performance and availability. Even brief database failures can now cost millions of dollars to companies.

To help businesses mitigate these problems and provide reliable uninterrupted service, DBA InfoPower developed a line of products that improve database performance, availability and manageability.

With DBAip's solution, enterprises can improve database performance, avoid costly downtime and other service interruptions, and make IT personnel more efficient in preventing problems and managing daily database operation.

DBAip's solutions should deliver immediate return on investment for their enterprise customers as they allow more efficient and effective use of customers' critical database resources.





# Welcome to Performance Explorer-i for Oracle

**Performance Explorer-I™** is a complete root cause analysis product that enables rapid discovery of database overload causes, replacing complex and time-consuming manual performance analysis and report generating products.

Unique analytical capabilities and interface of Performance Explorer-I provide you with a clear visualization and analysis of activity on your database. Snapshots Compare identifies system and load changes causing changes in performance on a single database or multiple databases, while Visual Analysis System vividly displays spikes and changes in key statistics and metrics.

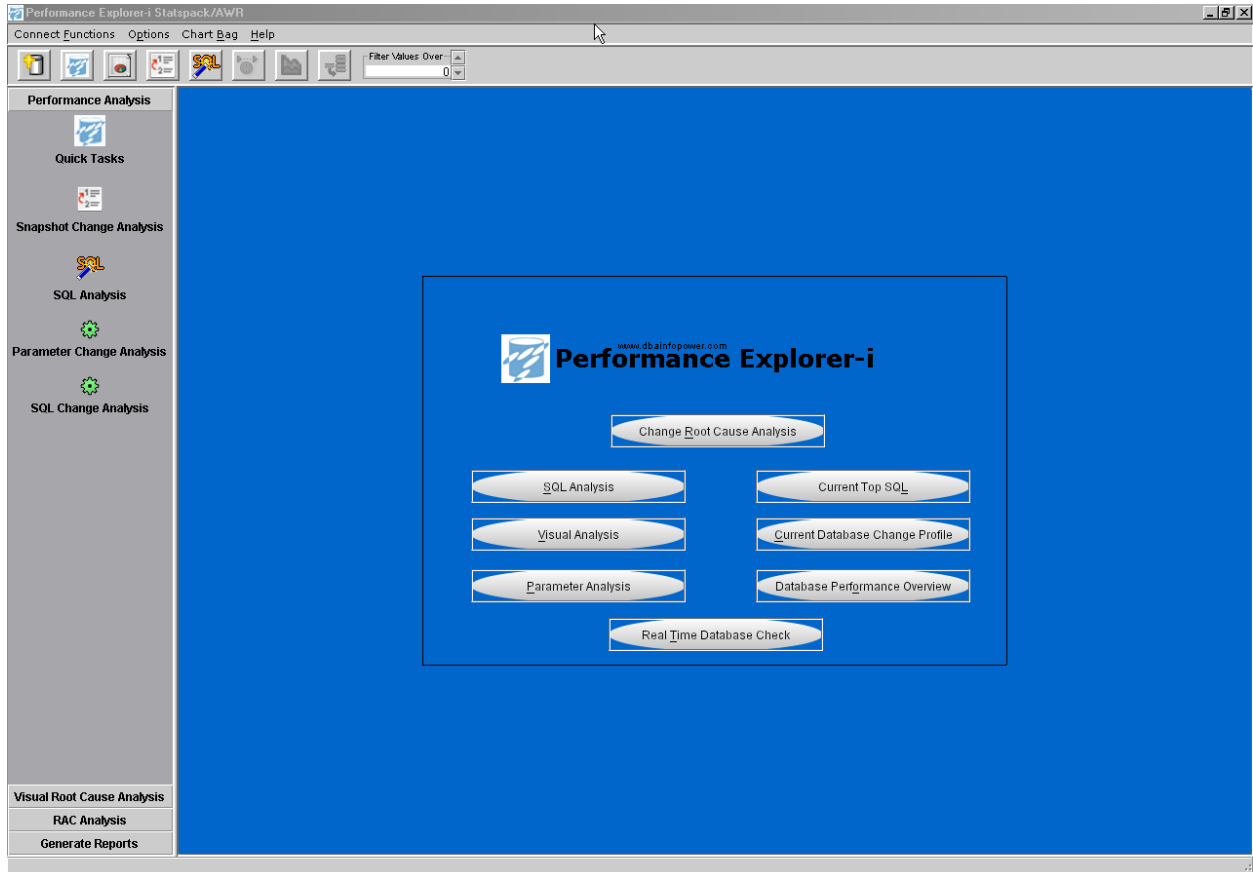
This chapter explains main features of Performance Explorer-i for Oracle.

In this chapter	Page
An overview of Performance Explorer-i for Oracle	7
Features of Performance Explorer-i for Oracle	15

# Overview of Performance Explorer-i for Oracle

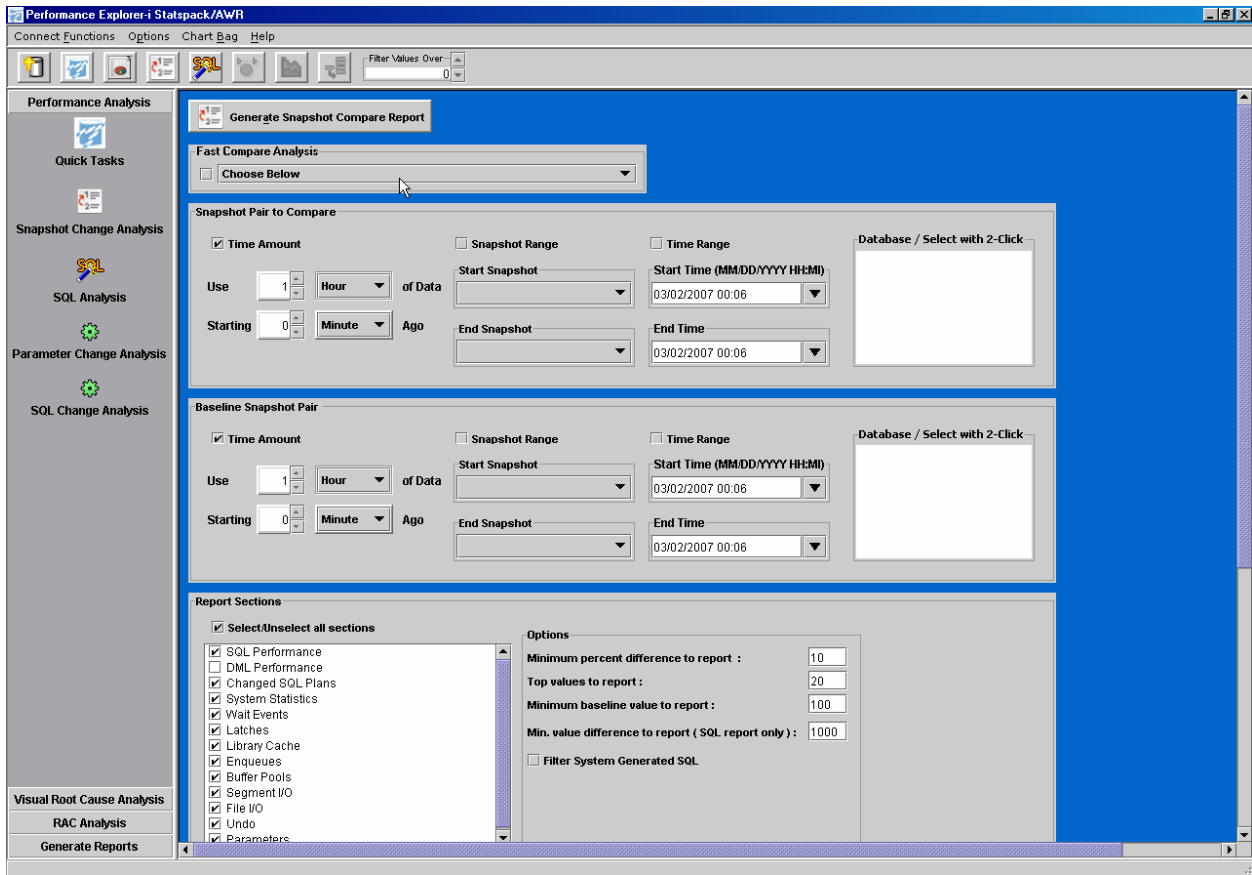
## Quick Tasks Panel

Quick Tasks Panel enables users to instantly obtain review of changes in database performance, compared to the previous week, identify most problematic SQL immediately impacting database and system performance, locate severe database bottlenecks in real-time, and receive access to most important and frequently used sections of Performance Explorer-i.



# Snapshot Change Analysis Section

Snapshot Change Analysis section enables users to receive detailed change profile of database performance, compared to the baseline period.



# SQL Analysis Section

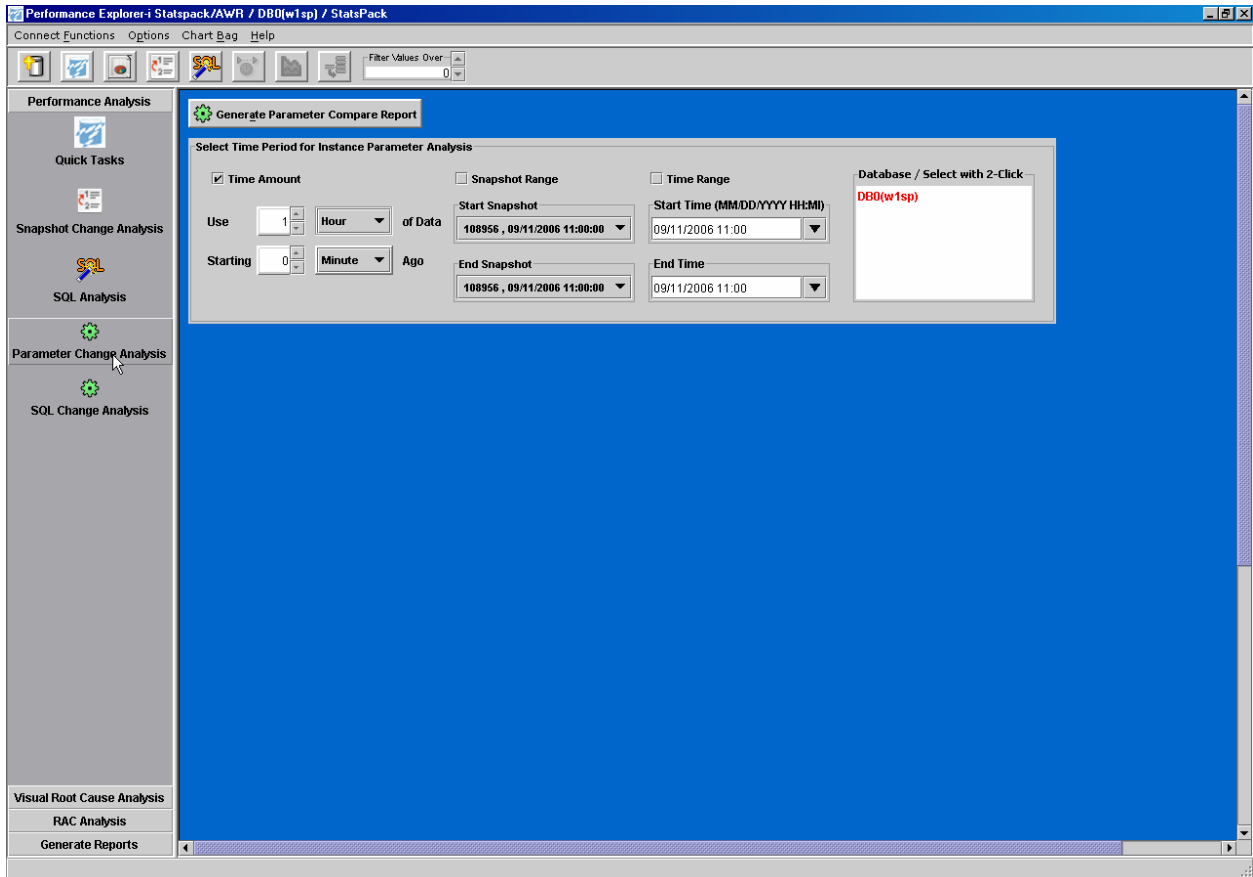
SQL Analysis provides users with detailed analysis and visualization of the most problematic SQL statements impacting database performance.

The screenshot displays the Oracle Performance Explorer interface for SQL Analysis. The main window shows a table of SQL statements with columns for Hash Value, SQL (Hold Mo...), BUFFER\_GETS, Total Executio..., Avg. Buffer Get..., and Avg. Disk Rea... The table lists several SQL statements, including SELECT TRX\_C..., begin RETURN..., declare be..., begin RETURN..., SELECT CONSI..., begin out\_ret..., SELECT RESO..., SELECT SUM(A..., SELECT RSP P..., SELECT\*+ IND..., DELETE FROM ..., SELECT RS.RA..., and BEGIN perfstat... The interface also includes a Search Panel, a Chart section with options for Split Chart, Calculate Trend (LR), and Smooth (Average) Chart Data Over, and a Set Time Periods for Visual Analysis section with options for Single Time Range and Cross-Time Range. The SQL Statistics panel on the right lists various metrics such as EXECUTIONS, BUFFER\_GETS, DISK\_READS, ROWS\_PROCESSED, SORTS, PARSE\_CALLS, LOADED\_VERSIONS, LOADS, INVALIDATIONS, VERSION\_COUNT, SHARABLE\_MEM, FETCHES, CPU\_TIME, and ELAPSED\_TIME.

Hash Value	SQL (Hold Mo...)	BUFFER_GETS	Total Executio...	Avg. Buffer Get...	Avg. Disk Rea...
286666270	SELECT TRX_C...	5,323,968	7,584	702	0
2009967854	begin RETURN...	3,182,084	876	3,733.505	1.78
2735175695	declare be ...	1,251,430	285	5,781.09	17.69
905505110	begin RETURN...	1,056,849	805	1,972.41	30.605
2172253855	SELECT CONSI...	656,212	277	2,951.905	12.13
4152144838	begin out_ret...	613,667	1,665	197.21	3.38
1258887367	SELECT RESO...	521,855	7,674	51.305	0
937558080	SELECT SUM(A...	496,809	554	282.605	0.04
1304046115	SELECT RSP P...	462,198	22,274	26	0
1471089065	SELECT*+ IND...	444,433	226	1,685.49	9.6
493477848	DELETE FROM ...	439,659	7,676	46.49	0
654547061	SELECT RS.RA...	416,027	23,036	21.205	0
2948768197	BEGIN perfstat...	360,044	4	92,748.453	1,626.28

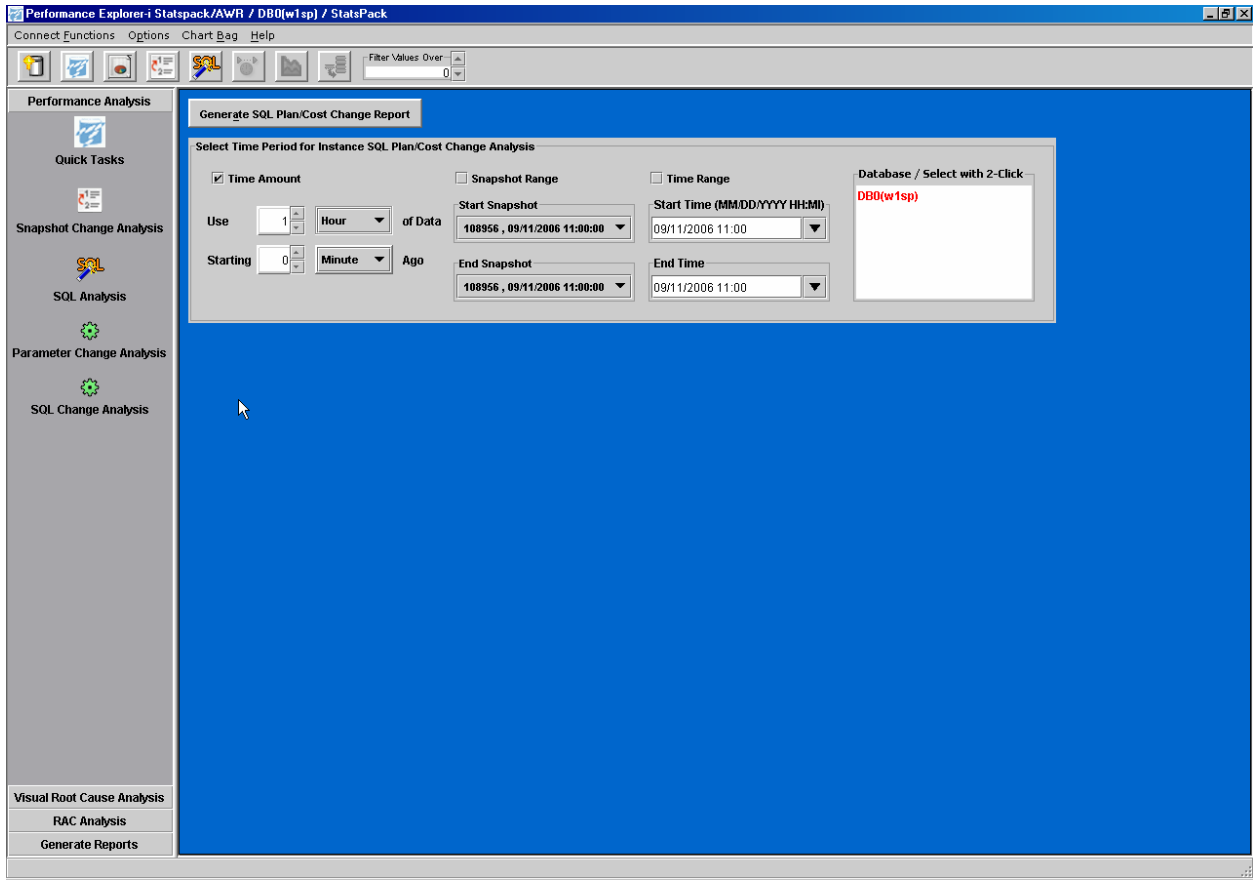
# Parameter Change Analysis Section

Parameter Change Analysis scans user provided time period and identifies any changes in database parameters made during that period by users or system.



# SQL Change Analysis Section

SQL Change Analysis scans user provided time period and identifies any changes in SQL statement execution plan of SQL Statement cost that happened during that period.



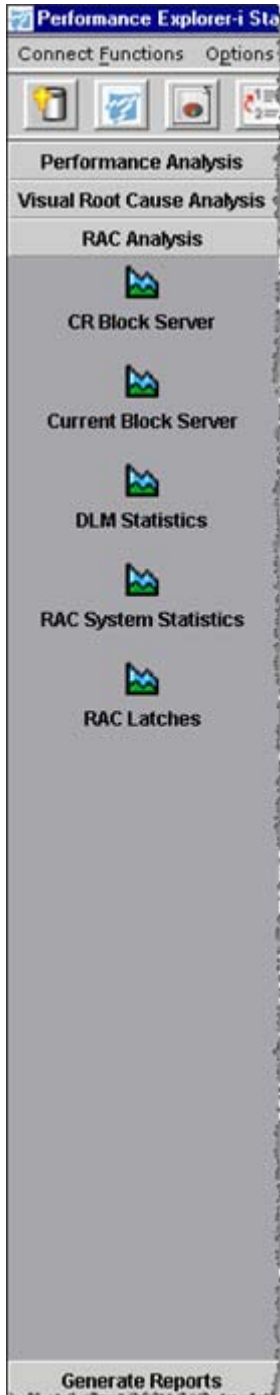
## Visual Root Cause Analysis Sections

Multiple Visual Root Cause Analysis sections enable users to easily analyze changes in database behavior by graphically displaying most important database performance statistics, wait events and other performance metrics, and allowing users to visually overlay graphs corresponding to different time periods.



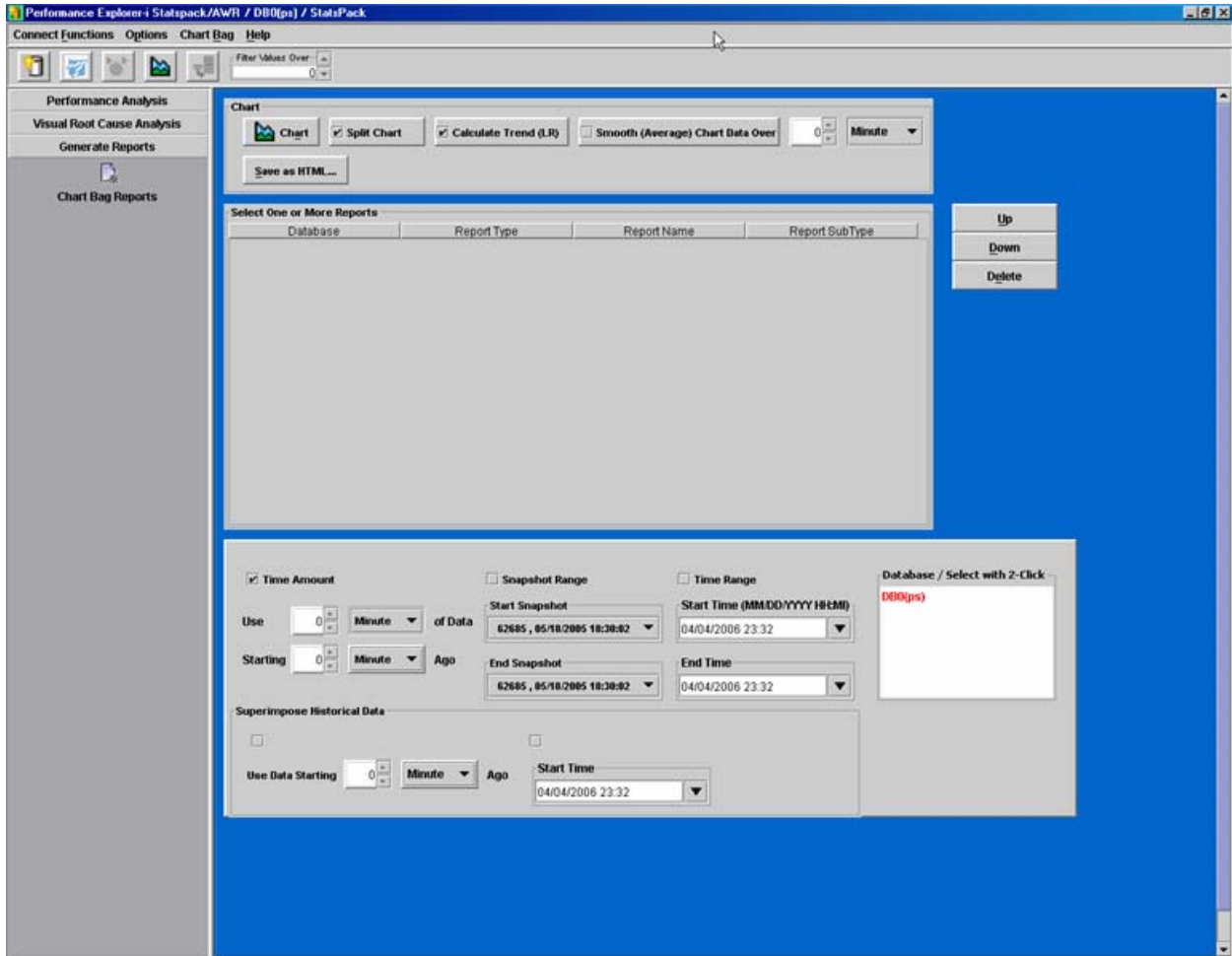
# RAC Analysis Sections

Multiple RAC Analysis sections enable users to easily analyze changes in RAC cluster behavior by graphically displaying most important RAC performance statistics and other performance metrics, and enables users to visually overlay graphs corresponding to different RAC nodes and time periods.



# Reports Section

Performance Explorer-i Reports Section provides users with access to ChartBag report generator that allows creation of performance reports as well as performance report templates both in interactive and batch modes



# Features of Performance Explorer

Performance Explorer-i is designed with ease of installation and simplicity of use in mind. Combination of powerful analytical capabilities and simplicity of operation makes Performance Explorer-i a tool of choice for complex production and development database environments.

## Features of Performance Explorer-i include:

- ◆ Easy installation
- ◆ Easy to use help system and manuals
- ◆ Extremely fast operation on vast arrays of collected data with no noticeable impact on system performance. Product uses only existing AWR/Statspack data. No data objects or stored procedures are created in the database.
- ◆ Root cause analysis of changes in database performance
- ◆ SQL performance analysis engine
- ◆ Visual Root Cause Analysis of database performance
- ◆ Visual Top Spike Analysis of database performance
- ◆ Discovery of changes in database parameters, as well as change timeframes
- ◆ Change root cause analysis and visual root cause analysis across multiple databases
- ◆ Change root cause analysis and visual root cause analysis across multiple database versions
- ◆ Change root cause analysis and visual root cause analysis across multiple RAC nodes
- ◆ Cross-time period analysis combining and displaying multiple time periods based on cross-time profile



# Getting started with Performance Explorer-i for Oracle

This chapter covers everything you need to know to start using Performance Explorer-i for Oracle. It explains how to:

- Install Performance Explorer-i
- Start Performance Explorer-i
- Connect to a database

In this chapter	Page
Installing Performance Explorer-i	16
Starting Performance Explorer-i	17
Connecting to an Oracle database	18

## Installation of Performance Explorer-i for Oracle

This procedure outlines how to install Performance Explorer-i on your workstation. Installation of Performance Explorer-i is very straightforward and should only take a few minutes to complete. Close all other applications before loading the CD into your PC. You can cancel installation at any time by pressing the **ESC** key or by clicking the **Cancel** button.

### To install Performance Explorer-i for Oracle

Insert the Performance Explorer-i CD in your CD-ROM drive or download Performance Explorer-i installer from the URL provided by a representative of DBA InfoPower, Inc.

Depending on your system configuration, installation program may start automatically, or you may have to start it yourself. If installer starts automatically, follow instructions on the screen to complete installation.

If the installation program does not start automatically when CD is inserted, select **Run** from the Windows Start menu.

Type **d:\pexplorer\_i\_401.exe** (where **d**: is the CD-ROM identifier and **401** is the version number), and click **OK**.

Follow the instructions displayed on the screen.

By default, Performance Explorer-i for Oracle is installed in the following directory, though you can choose a different installation path.

**C:\Program Files\ DBAip Performance Explorer-i**

## **Multiple installations of Performance Explorer-i for Oracle**

Advanced Performance Explorer-i installer allows you to install and separately upgrade multiple versions of Performance Explorer-i for Oracle on your computer.

## **Authorizing Performance Explorer-i for Oracle**

If you are a new user of Performance Explorer-i for Oracle, you will have a trial period in order to test the product. When the trial period expires, Performance Explorer-i for Oracle will stop functioning.

To maintain full functionality, contact your DBA InfoPower, Inc. representative for information on obtaining a full licensed version. For further information see “Product authorization errors”.

## **Uninstalling Performance Explorer-i for Oracle**

To uninstall Performance Explorer-i for Oracle, use the **Add/Remove Programs** option in the Windows **Control Panel** in the **Start** menu.

## **Starting Performance Explorer-i for Oracle**

In order to connect to a database, you need to complete the following steps:

1. Launch the application
2. Connect to the desired database or multiple databases

Performance Explorer-i for Oracle can be launched in one of two ways: directly, or from DBA HeartBeat Console. The simplest way to get started is to launch Performance Explorer-i directly.

## **Launching Performance Explorer-i Directly**

Performance Explorer-i can be launched directly from the Windows Start menu. Appropriate Performance Explorer-i can be found on the Windows Start menu under:

1. **DBAip Performance Explorer-i > Performance Explorer-i Statspack-AWR, 10g JDBC Driver**  
Or
2. **DBAip Performance Explorer-i > Performance Explorer-i Statspack-AWR, 9i JDBC Driver**  
Or
3. **DBAip Performance Explorer-i > Performance Explorer-i Statspack-AWR, 8i JDBC Driver**

## **Launching From DBA HeartBeat Console**

When DBA HeartBeat Console is available, you can also launch Performance Explorer-i by

selecting **Root Cause Analysis** option on the DBA HeartBeat Console.

## Connecting to Database

Use the steps below to connect to Oracle database

- 1) From the “**Connect Functions**” menu, choose “**Connect...**”
- 2) In the “Database TNS Alias or thin URL” text box enter database tnsnames alias or JDBC database URL. Click on “jdbc:oracle:thin:@<hostname>:<port>:<SID>” label to paste it into the text box.
- 3) In a RAC environment, set instance number to the instance number required for the analysis session (default instance number is 1)
- 4) In the “Statspack/AWR user” text box, enter name of the user authorized to read statspack tables or AWR performance views (empty statspack username defaults to “perfstat”)
- 5) In the “Password” text box, enter password of the user authorized to read statspack tables or AWR performance views (empty password defaults to be the same as the username)



# 4

## Common Control Elements of Performance Explorer-i for Oracle

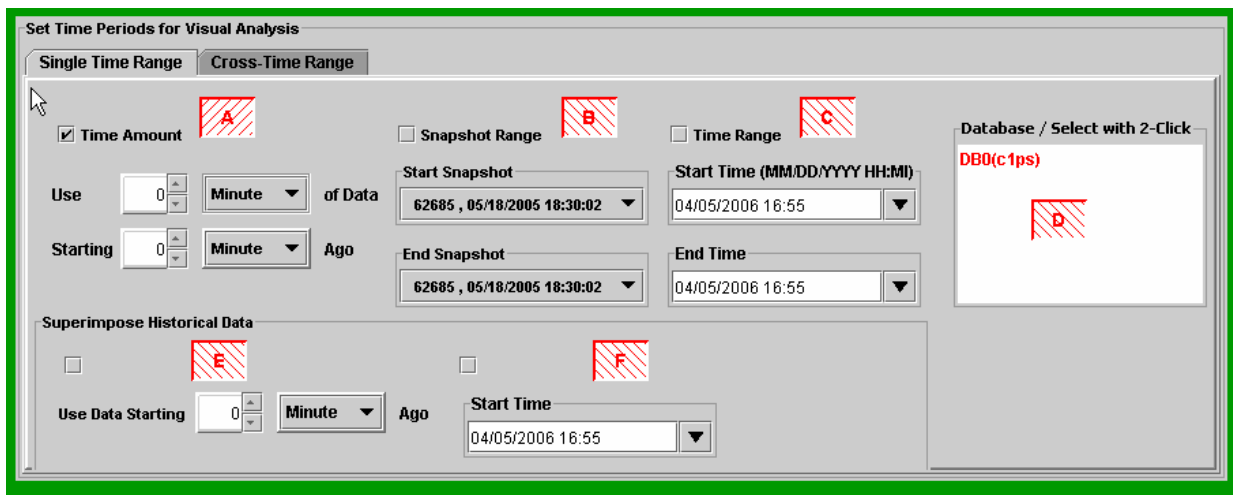
<b>In this chapter</b>	<b>Page</b>
Performance Explorer-i Common Control Elements	20
Timeline control sub-panel	20
Cross-Time control sub-panel	21
Chart Control sub-panel	22
Top Search control sub-panel	23
Statistic Selection control sub-panel	24

## Performance Explorer-i Common Control Elements

This section describes common user interface control features of the Performance Explorer-i for Oracle. The controls can be of the following kinds:

### Timeline control sub-panel:

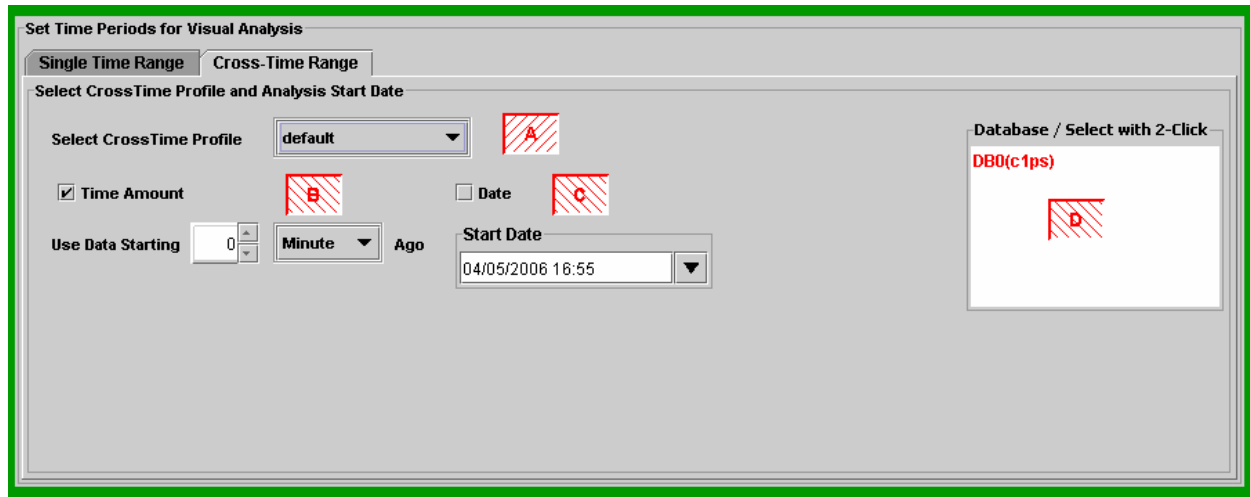
Timeline control sub-panel allows users to set time periods of the root cause analysis. Timeline control sub-panel consist of the components listed below:



- A) **Time Amount block** – set time period for analysis in terms of amount of time. Optionally, time offset from a point in the past can be used.  
**For Example:** Use “1 Day” of data or “Use 1 Day of Data starting 1 Week Ago”
- B) **Snapshot Range block** – set time period for analysis in terms of starting and ending statspack/AWR snapshots  
**For Example:** Snapshot Range between snapshots 62640 and 62678
- C) **Time Range block** – set time period for analysis in terms of starting and ending timestamps  
**For Example:** Start Time is set to 04/05/2006 14:00 and 04/05/2006 16:15
- D) **Database List** – displays list of currently connected databases. User can make database active within a panel by double-clicking it with the pointing device. To make database active across all panels, user should triple-click selected database with the pointing device.
- E) **Superimpose Time Amount Block** – select time period to superimpose with the one selected in A) - C) by indicating how far in the past the corresponding time period should be taken from.  
**For Example:** Use Superimpose Data starting “1 Week” ago
- F) **Superimpose Start Time Block** – select time period to superimpose with the one selected in A) - C) by indicating the starting timestamp of that period  
**For Example:** Superimpose Time is set to 03/29/2006 14:00

## Cross-Time Range Control Sub-panel

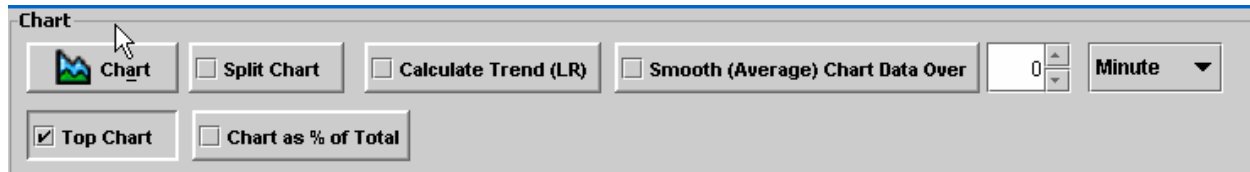
Cross-Time Range control sub-panel allows users to set cross-time profile and time periods needed for the cross-time analysis. For more information on cross-time analysis, see “Tuning Your Database” section. Cross-Time Range control sub-panel consists of the following components:



- A) **Cross-Time Profile Selection** – allows user to select cross-time profile for analysis
- B) **Time Amount Block** – set cross-time analysis time by using amount of time to be jumped in the past  
**For Example:** Use Data starting “1 Day” ago
- C) **Start Time Block** – set cross-time analysis time by using timestamp  
**For Example:** Time is set to 03/29/2006 14:00
- D) **Database List** – displays list of currently connected databases. User can make database active within a panel by double-clicking it with the pointing device. To make database active across all panels user should triple-click selected database with the pointing device.

## Chart Control sub-panel

Chart Control sub-panel allows users to set a parameter for visual analysis and create graph that contains analysis results. For detailed information on visual root cause analysis see “Tuning Your Database” section. Chart Control sub-panel consists of the following components:



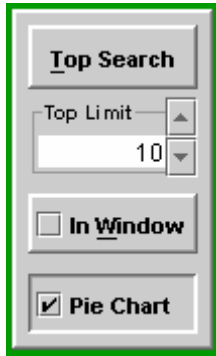
- A) **Chart button** - creates visual analysis graph
- B) **Split Chart checkbox** – combines analysis of multiple performance metrics on a single graph or splits it into separate graphs (one per performance metric)
- C) **Calculate Trend checkbox** – creates linear regression model based on performance metrics data. Resulting Trend line provides simple and effective way to understand headroom dynamics and predicts capacity requirements.
- D) **Smooth (Average) Chart Data Over block** – averages performance metrics data over the selected time period. This option is especially useful during visual analysis of long time periods, since it eliminates visual clutter introduced by large numbers of data points
- E) **“Top Chart” checkbox** – creates top spike analysis chart of performance metric statistic.  

**For Example:** Top Chart Visual Analysis of SQL “BUFFER GETS“ displays results of analysis as continuous visual display of N-top BUFFER\_GETS SQL Statements for every time point of the provided time range.
- F) **“Chart as % or Total” checkbox** – displays results of analysis as percent of overall value of performance metric statistic.  

**For Example:** Create Visual Analysis of “TOTAL\_WAITS “ of “db file sequential read”, display results of analysis as % of overall “TOTAL\_WAITS” time across all events
- G) **“Save as HTML” checkbox** (not shown) – option specific to ChartBag reports – saves ChartBag report as HTML file. For more information on ChartBag please see “Using Performance Explorer-i for Reporting” section.

## Top Search control sub-panel

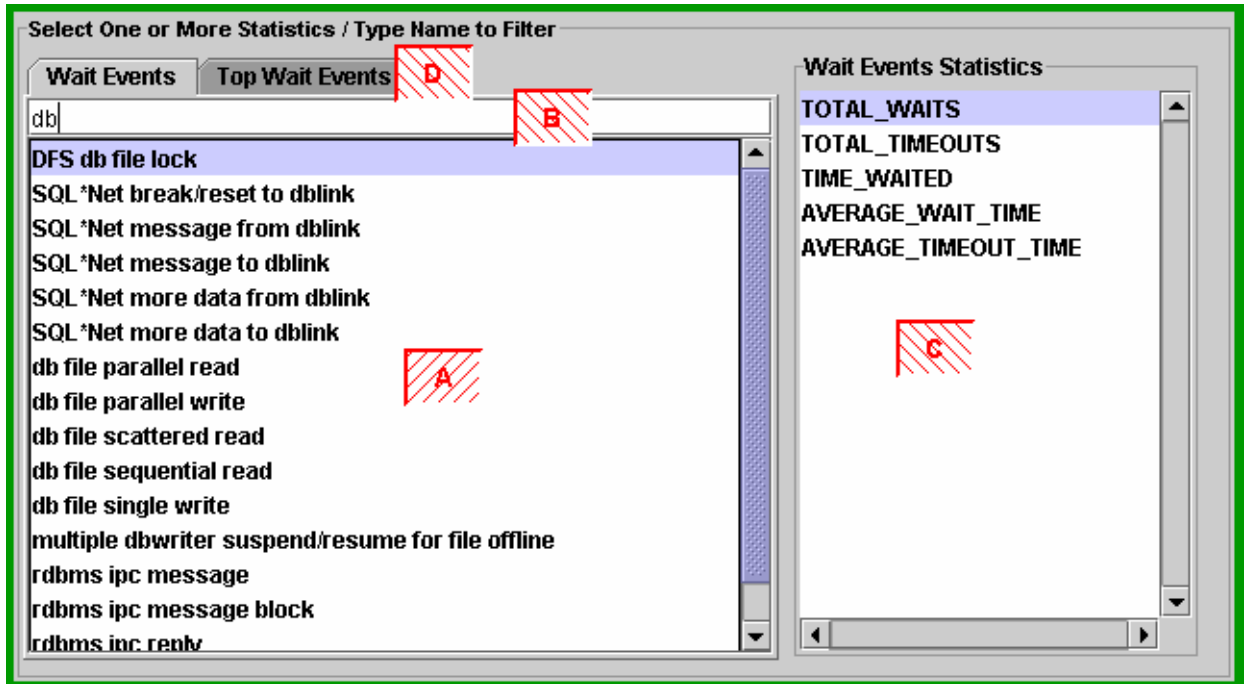
Top Search control sub-panel allows users to set parameter for Top Search of the database performance metrics. Top Search allows a user to display and monitor top database metrics (e.g., wait events), i.e., metrics that have predominant impact on database performance. Top Search control sub-panel consists of the following components:



- A) **Top Search** Button - executes top search
- B) **Top Limit** spinner – sets amount of top search results elements to return
- C) **In Window** checkbox – opens Top Search result in a separate window. This option is helpful when multiple top search results are needed to be created and reviewed at the same time
- D) **Pie Chart** checkbox – create Pie Chart Graph in addition to the table based Top Search report

## Statistic Selection control sub-panel

Statistic Selection control sub-panel is used by all Visual Root Cause Analysis sections. It is customized for each performance metric area. Statistic Selection control sub-panel consists of the following components:



- A) **Performance Metrics List, Tree or Table** (depending on section) – display list of performance metrics, such as system statistics, wait events, latches, I/O components, etc.
- B) **Filter textbox** – allows to filter master list A) based on the typed-in pattern
- C) **List of Performance Metric Statistic** – list of statistics related to given section of performance metrics
- D) **Top Result Table** – shows results of the Top Search



# 5

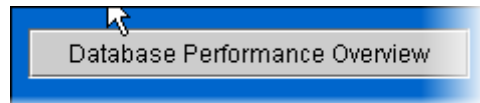
## Using Performance Explorer-i for Oracle

When you use Performance Explorer-i for Oracle to diagnose an Oracle database, you can:

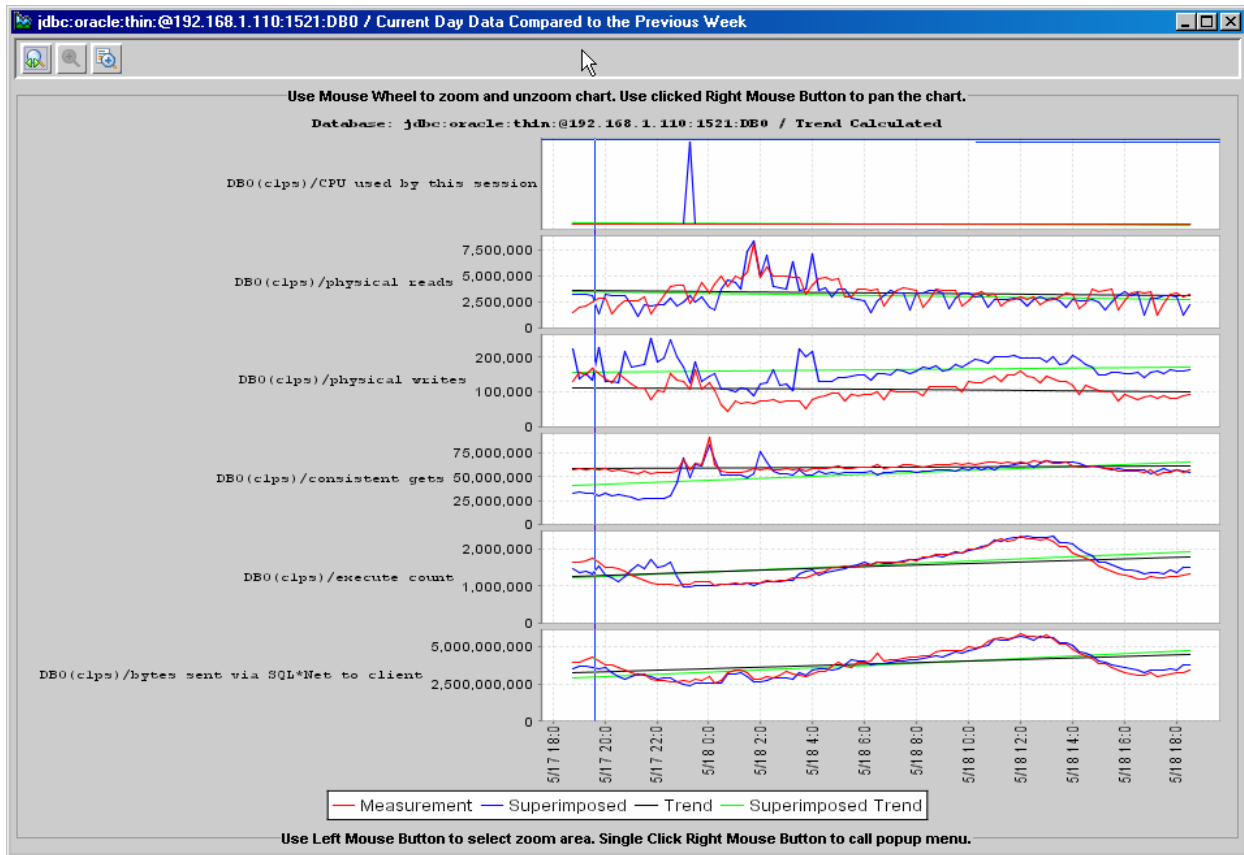
- View status of the connected database using Performance Explorer-i **Quick Tasks** page.
- View performance change analysis created by Performance Explorer-i.
- View detailed visual performance change analysis of the connected database in various Performance Explorer-i **sections**.

In this chapter	Page
Performance Explorer-i Quick Tasks Panel	26
Performance Explorer-i Snapshot Change Analysis	29
Performance Explorer-i SQL Analysis	32
Performance Explorer-i Parameter Change Analysis	36
Performance Explorer-i Visual Root Cause Analysis	38
Performance Explorer-i ChartBag Reports	40
Performance Explorer-i Cross Database Analysis	42
Performance Explorer-i Cross Time Analysis	43
Using Performance Explorer-i to analyze and troubleshoot your database	46
Using Performance Explorer-i for performance reporting	57
Using Performance Explorer-i to create performance and headroom web site	58
Using Performance Explorer-i in secure environment (SSH tunneling)	60

## Performance Explorer-i Quick Tasks



Database performance overview task provides users with an overview of the most important database performance metrics superimposed with the historical data from the previous week. This information enables immediate discovery of changes in database performance.



## Current Database Change Profile

Current Database Change Profile task makes possible immediate change root cause analysis by comparing most current data with the previously collected snapshots. This task enables a user to determine causes of the most recent changes in database performance – e.g., in a case of new code deployment or upgrade.

Comparing Snapshots 62684 - 62685 ( 2005-05-18 18:15:03.0 to 2005-05-18 18:30:02.0 )  
With 62683 - 62684 ( 2005-05-18 18:00:02.0 to 2005-05-18 18:15:03.0 )

Changed/New Instance Parameters Section

Param	Name	End Value	Start Value
Baseline Changed Parameters	sort_area_size	8120000	4096000
Compare Changed Parameters	sort_area_size	4096000	8120000
Compare/Baseline Changed Parameters	sort_area_size	4096000	8120000

Changed Execution Plans SQL Section

<<<!!!SQL Execution Plan Change Tracking is not provided for 8i databases!!!>>>

SQL Performance Section

SQL (EXECUTIONS)

Hash Value	Difference (Pct)	Compared Value	Baseline Value	Compared/Exec	Baseline/Exec	Difference/Exec	SQL Text
754324216	151,500	1,515	0	1	0	1	select ad_id
957695179	21,543	1,515	7	1	1	0	select id fro
3316429131	63	89,316	54,697	1	1	0	SELECT /* Sal
2292834205	20	15,843	13,166	1	1	0	SELECT /* Sal
1047666738	18	243,883	206,267	1	1	0	SELECT /* Act

SQL (BUFFER\_GETS)

Hash Value	Difference (Pct)	Compared Value	Baseline Value	Compared/Exec	Baseline/Exec	Difference/Exec	SQL Text
1942515845	822,400	8,224	0	8,224	0	8,224	declare cur
2117115962	820,200	8,202	0	8,202	0	8,202	SELECT ROWID
3518605083	398,800	3,988	0	199	0	199	SELECT /* Sal
754324216	303,200	3,032	0	2	0	2	select ad_id
957695179	23,954	6,735	28	4	4	0	select id fro
1845441572	3,441	4,143	117	82	58	24	UPDATE /* Sal
1032544972	894	7,933	798	146	44	102	delete /* Rem
88123355	319	3,720	887	413	177	236	SELECT /* Off
875037822	212	6,042	1,935	154	138	16	SELECT /* Tre
2517784366	186	1,853	647	1,853	0	1,853	select /* Get
1102578034	179	1,821	653	11	11	0	INSERT INTO B
160808594	139	3,598	1,504	58	57	1	SELECT /* Tre
3941715755	137	2,208	930	39	15	24	delete /* Del
661593930	114	2,787	1,300	146	76	70	SELECT /* Tre
1574905954	100	450,666	225,390	225,333	225,390	-57	SELECT /*
1565868173	100	450,604	225,332	225,302	225,332	-30	SELECT /*

## Visual Analysis

Visual Analysis switches the user to the Visual Root Cause Analysis Section

## Current Top SQL

Current Top SQL generates report (in a form of table and a pie Chart) of most recent top resource-consuming SQL statements, taking into account all major SQL statistics (like BUFFER GETS, DISK READS, EXECUTIONS, CPU TIME, etc).

SQL Analysis

SQL Analysis switches the user to the SQL Analysis Section

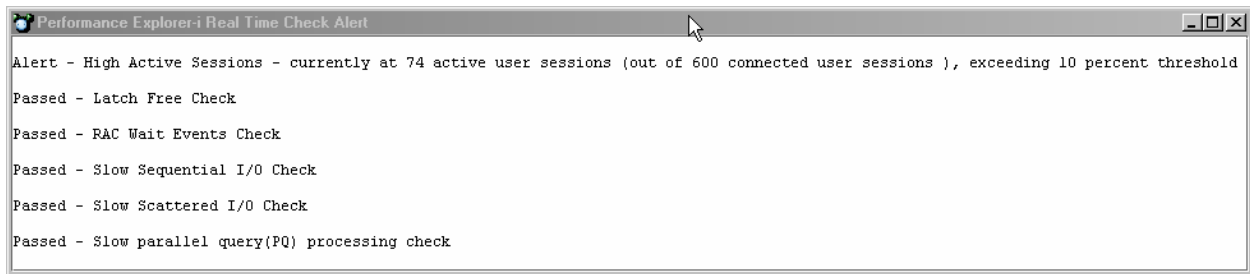
Parameter Analysis

Parameter Analysis switches the user to the Parameter Analysis Section

Real Time Database Check

Real Time Database Check determines immediate threats to database stability, as well as potential performance bottlenecks. Some of the performed checks are:

- 1) Check for excessive number of active sessions
- 2) Detection of Latch Connection
- 3) RAC related resource contention
- 4) Slow I/O (random and sequential data access)
- 5) Excessive query parallelism in the database

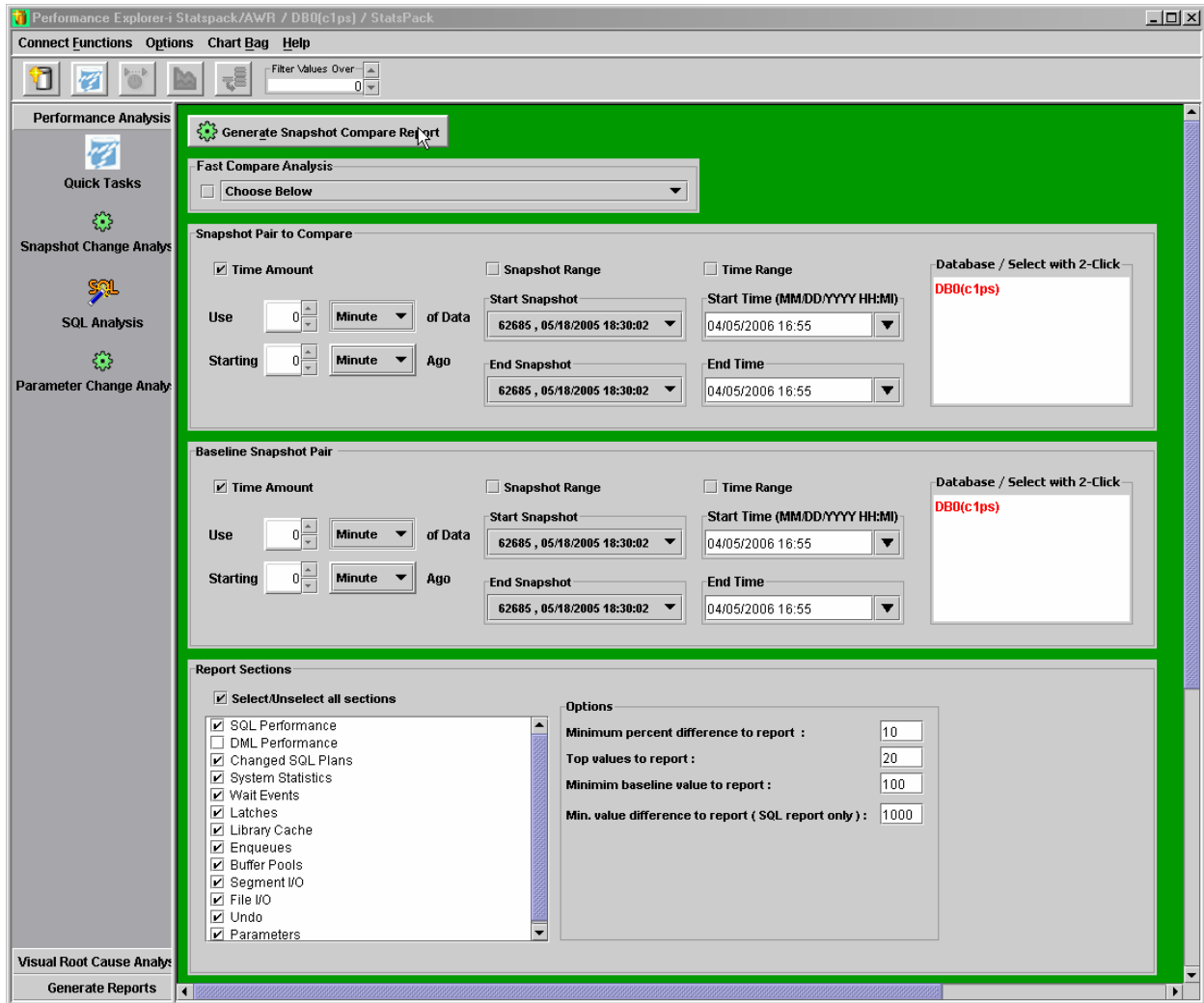


## Performance Explorer-i Snapshot Change Analysis

Snapshot Change Analysis report eliminates the need for painfully tedious eyeballing of statspack/AWR reports in the situations when ability to resolve a problem about to bring down the database is critical. Detailed reports show changed SQL statistics, newly introduced SQL, changed SQL plans, system statistics, wait events, latches, as well as many other performance metrics that crossed the changed threshold and can potentially be the root cause of negative changes in database availability and performance.

### Snapshot Change Analysis Panel

The following screen shows main features of the Snapshot Change Analysis Panel



Following are the main elements of the panel:

**Compare Snapshot Timeline Sub-panel** allows users to set period of time to perform change root cause analysis.

**Snapshot Pair to Compare**

Time Amount       Snapshot Range       Time Range

Use  Minute of Data      Start Snapshot: 62685 , 05/18/2005 18:30:02      Start Time (MM/DD/YYYY HH:MI): 04/05/2006 16:55

Starting  Minute Ago      End Snapshot: 62685 , 05/18/2005 18:30:02      End Time: 04/05/2006 16:55

Database / Select with 2-Click: DB0(c1ps)

**Baseline Snapshot Timeline Sub-panel** allows users to set period of time to be used as a baseline for the change root cause analysis.

**Baseline Snapshot Pair**

Time Amount       Snapshot Range       Time Range

Use  Minute of Data      Start Snapshot: 62685 , 05/18/2005 18:30:02      Start Time (MM/DD/YYYY HH:MI): 04/05/2006 16:55

Starting  Minute Ago      End Snapshot: 62685 , 05/18/2005 18:30:02      End Time: 04/05/2006 16:55

Database / Select with 2-Click: DB0(c1ps)

**Fast Compare Sub-panel** allows users to perform fast root cause analysis based on comparison of most recently collected performance data with the preset past time periods

**Fast Compare Analysis**

Choose Below

Choose Below

- SPNOW - Compare latest snapshot pair with the previous one
- SPHOUR - Compare latest snapshot pair with the one from hour ago
- SPDAY - Compare latest snapshot pair with the one from day ago
- SPWEEK - Compare latest snapshot pair with the one from week ago
- SPMON - Compare latest snapshot pair with the one from month ago

**Report Sections Sub-panel** allows users to select areas of database performance for investigation and set required+ reporting thresholds

The screenshot shows a window titled "Report Sections". On the left, there is a list of sections with checkboxes. The first checkbox is checked and labeled "Select/Unselect all sections". Below it, the following sections are listed with checked boxes: SQL Performance, Changed SQL Plans, System Statistics, Wait Events, Latches, Library Cache, Enqueues, Buffer Pools, Segment I/O, File I/O, Undo, and Parameters. The "DML Performance" checkbox is unchecked. On the right, there is an "Options" section with four input fields: "Minimum percent difference to report" (value: 10), "Top values to report" (value: 20), "Minimum baseline value to report" (value: 100), and "Min. value difference to report (SQL report only)" (value: 1000).

**Report options include:**

- A) Minimum percent difference to report
- B) Number of top values to report
- C) Minimum baseline value to report
- D) Minimum value difference to report (for SQL analysis only)

## Performance Explorer-i SQL Analysis

SQL Analysis enables finding top resource-consuming SQL statements and allows a user to create visual analysis to identify exact timeline of their impact on database performance. User can:

- Investigate top SQL based on SQL statistics or SQL statistic ratios;
- Aggregate groups of SQL for group visual analysis;
- Overlay current data with historical information;
- Perform cross-time analysis for selected SQL statements;
- Perform execution plan analysis for selected SQL statements.

### SQL Analysis Panel

The following screen shows main features of the SQL Analysis Panel

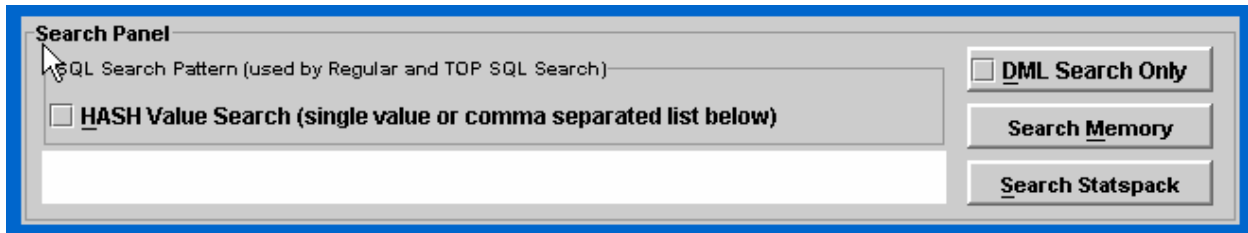
The screenshot displays the SQL Analysis Panel interface with several key components:

- Chart Panel:** Includes options for 'Chart', 'Split Chart', 'Calculate Trend (LR)', 'Smooth (Average) Chart Data Over', and a time interval set to 'Minute'.
- Search Panel:** Features a search pattern input field, a 'HASH Value Search' checkbox, and buttons for 'Search Memory' and 'Search Statspack'.
- Aggregate Selected SQL Under Name:** A text input field for grouping SQL statements.
- SQL (Ctrl-C to Copy) / Select One or More Statements to Chart or Explain:** A table listing SQL statements with their performance metrics.
- SQL Statistics:** A list of performance metrics such as EXECUTIONS, BUFFER\_GETS, DISK\_READS, ROWS\_PROCESSED, SORTS, PARSE\_CALLS, LOADED\_VERSIONS, LOADS, INVALIDATIONS, VERSION\_COUNT, SHARABLE\_MEM, BUFFER\_GETS/EXECUTIONS, DISK\_READS/EXECUTIONS, and SORTS/EXECUTIONS.
- Set Time Periods for Visual Analysis:** A section for configuring time ranges, including 'Single Time Range' and 'Cross-Time Range' options, with fields for 'Start Snapshot', 'End Snapshot', 'Start Time', and 'End Time'.

Hash Value	SQL (Hold Mouse O...	Top Value	Total Executi...	Avg. Buffer ...	Avg. Disk R...
2472647753	SELECT /* GetEOAlte...	22,341,748	600	193.53	0.23
711985649	SELECT /*+ index(b) *...	7,634,604	58	204.165	0
2164547437	select /*...	4,158,199	39	3,139.93	1.51
2292834205	SELECT /* SaleAggre...	2,990,812	15,843	8.58	4.52
2518134255	SELECT /* SaleAggre...	2,905,371	165,805	-0.805	0.35
600307448	SELECT /* SaleAggre...	2,316,597	150,841	0.59	0.56
3710846448	SELECT A1_C0 C0,A1...	2,122,947	148	-1,995.815	-2,031.26
1047666738	SELECT /* AttrInfoMap...	1,598,289	243,893	0.07	-0.46
2057893113	SELECT /* SaleAggre...	1,531,832	18,125	0.045	1.89
3103766504	SELECT /* OfferMap.F...	927,364	91,615	9.66	0.6
3316429131	SELECT /* SaleAggre...	744,978	89,316	0.685	0.88
1574905954	SELECT /*+ index(...	450,666	2	30,929.365	17.98
1565868173	SELECT /*+ index(...	450,604	2	31,302.805	219.755

Following are the main elements of the SQL Analysis panel:

- A) **Chart Panel** (see chart panel description in the “Common Control Elements” chapter)
- B) **Search Panel** – allows a user to search SQL (both in database memory and in statspack/AWR)



**Search Panel** is used by both SQL Search and Top SQL search modules.

SQL Search can be accomplished by providing search pattern or comma separated list of SQL Hash Ids and SQL Ids (in Oracle 10g+). For Hash Value Search, corresponding checkbox needs to be selected.

**DML Search Only** filter is used to limit search to DML (Insert/Select/Update/Merge) statements only.

- C) **Aggregate Select SQL** block allows user to group multiple SQL statement together for visual analysis

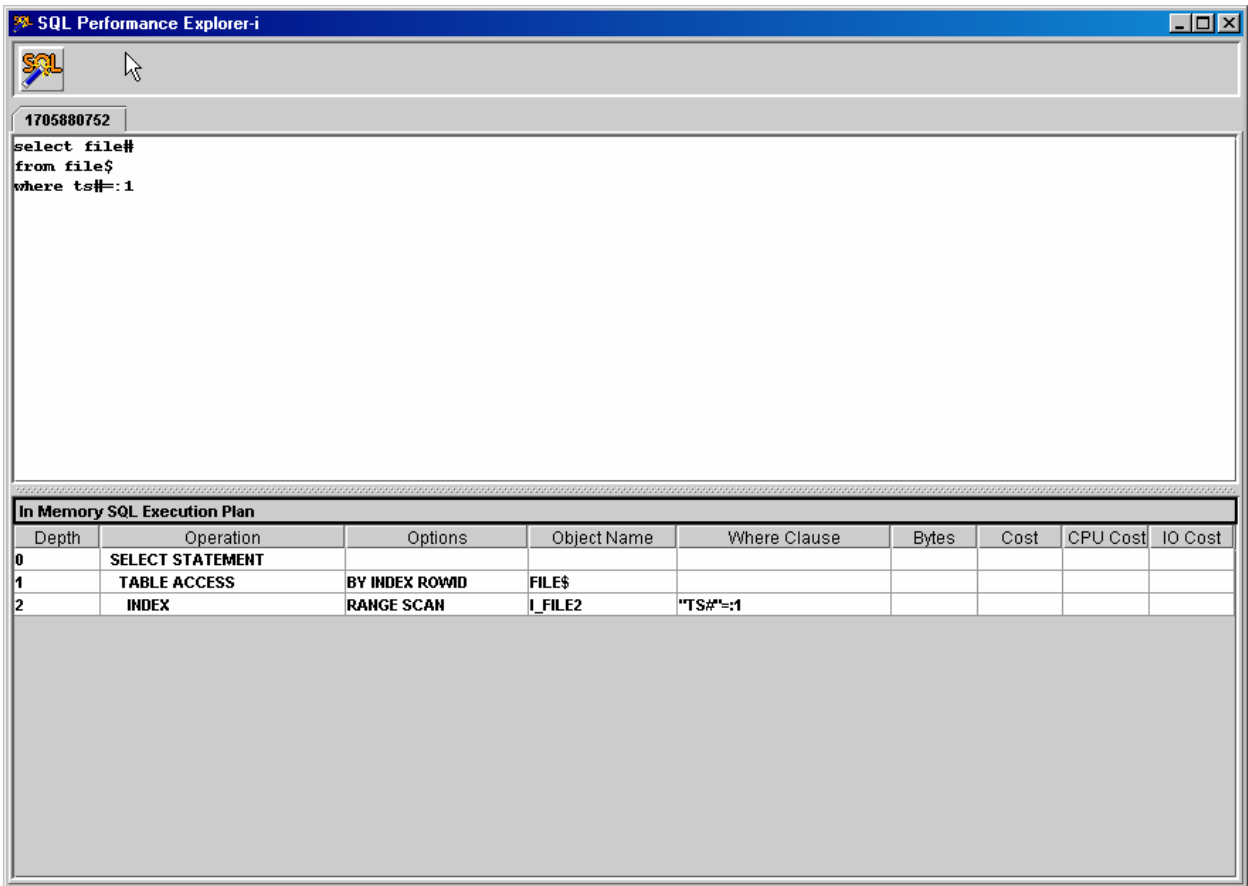


- D) **Explain Selected SQL** - generates execution plan for the selected SQL Statement



Created SQL execution plan presentation features “where clause” per explain plan step and separation of cost between CPU cost and I/O cost

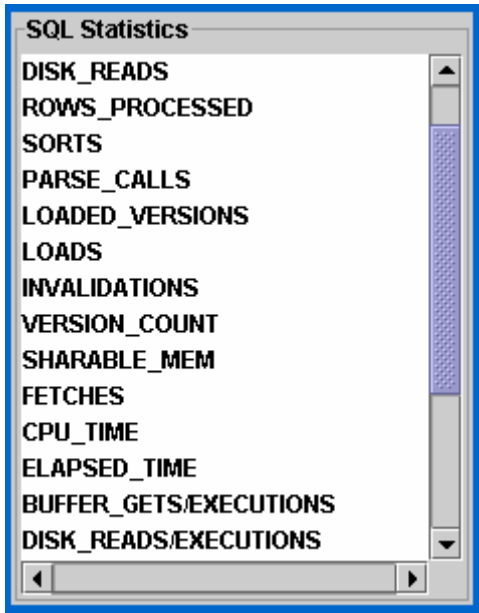
Initially SQL Explain displays plans recorded by Statspack/AWR. If such data is not recorded, explain plan is executed and resulting plan is shown to the user. User can modify SQL and re-explain statements as needed.



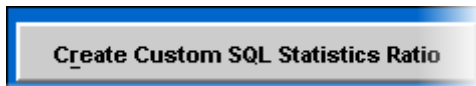
E) **SQL Statements** table – allows users to select a list of SQL statements to investigate. The SQL Statements table contains the original SQL statements as well as major performance statistics and ratios (such as Total SQL Executions, average ratios of buffer gets to executions and disk reads per execution). For the Top Search results, SQL Statements table contains top search value as well (under “Top Value” header). In order to see the whole SQL statement, the user needs to hold the mouse over the SQL column (mouse-over).

SQL (Ctrl-C to Copy) / Select One or More Statements to Chart or Explain					
Hash Value	SQL (Hold Mo...	Top Value	Total Executio...	Avg. Buffer Get...	Avg. Disk Rea...
600307448	SELECT /* Sale...	150,841	150,841	0.59	0.56
3103766504	SELECT /* Offer...	91,615	91,615	9.66	0.6
3316429131	SELECT /* Sale...	89,316	89,316	0.685	0.88
1644131110	SELECT /* Sale...	58,686	58,686	1.97	0.3
1477343896	SELECT /* Tran...	34,776	34,776	1.03	0.03
3549906293	select TO_CHA...	24,145	24,145	1.63	0
3814522649	SELECT /* Offer...	18,247	18,247	0.41	0.17
2057693113	SELECT /* Sale...	18,125	18,125	0.045	1.89
2292834205	SELECT /* Sale...	15,843	15,843	8.58	4.52
3977017273	SELECT /* Sale...	15,337	15,337	5	0.02
1168271026	update "SPLEX5...	14,505	14,505	8.39	0
3717435990	SELECT /* Sale...	12,960	12,960	13.305	0.42
2057700606	SELECT /* Sale...	12,611	12,611	16.56	0.65

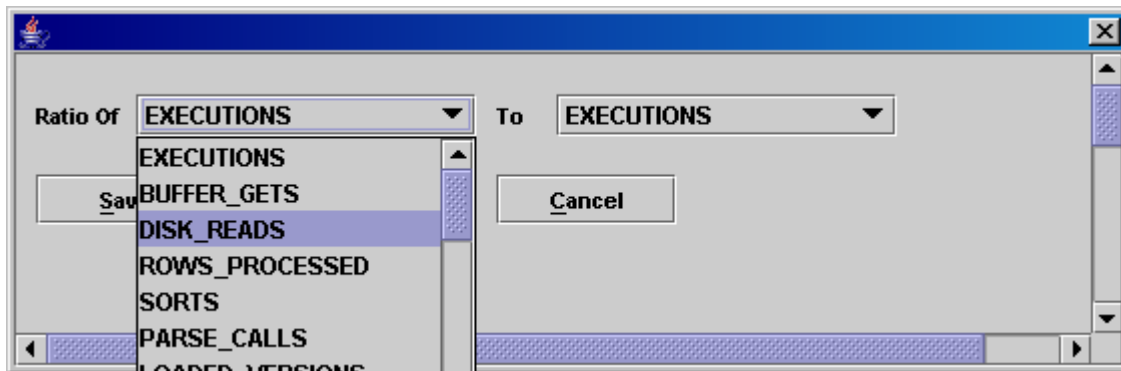
- F) **SQL Statistics** list allows users to select a list of SQL statistics to investigate. The list contains both the original SQL statistics, as well as pre-defined and user-defined SQL statistic ratios



- G) **Create Custom SQL Statistics Ratio** – allows users to define custom ratios, such as CPU\_TIME/PARSE\_CALLS, etc.



**Create Custom SQL Statistics** opens the following dialog box allowing the user to define custom SQL statistic ratio.



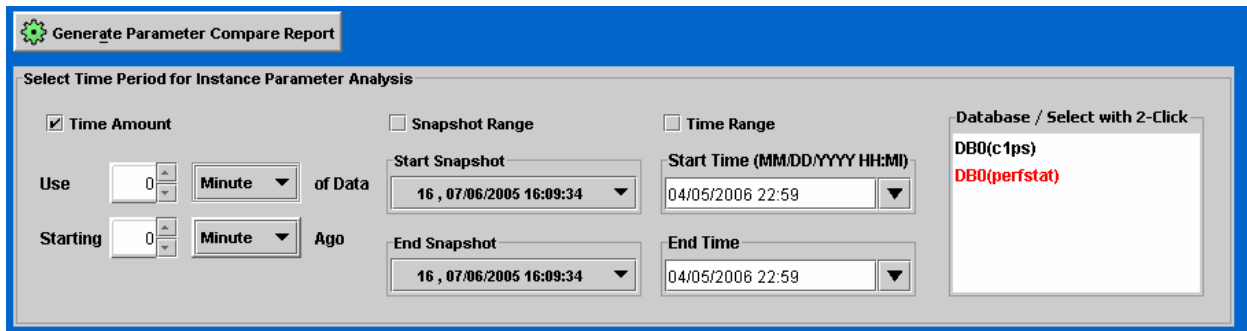
- H) **Top Search** panel (see Top Search panel description in the “Common Control Elements” chapter)
- I) **Timeline and Cross-Time Panels** (see Timeline control sub-panel and Cross-Time control sub-panel in the “Common Control Elements” chapter)

## Performance Explorer-i Parameter Change Analysis

Parameter Change Analysis enables finding the list of database parameters that have changed within a user provided time interval, as well as exact timeline for every change. Parameter Change Analysis helps a user to determine if a change in database performance may be correlated to changes in database parameters.

### Parameter Change Analysis Panel

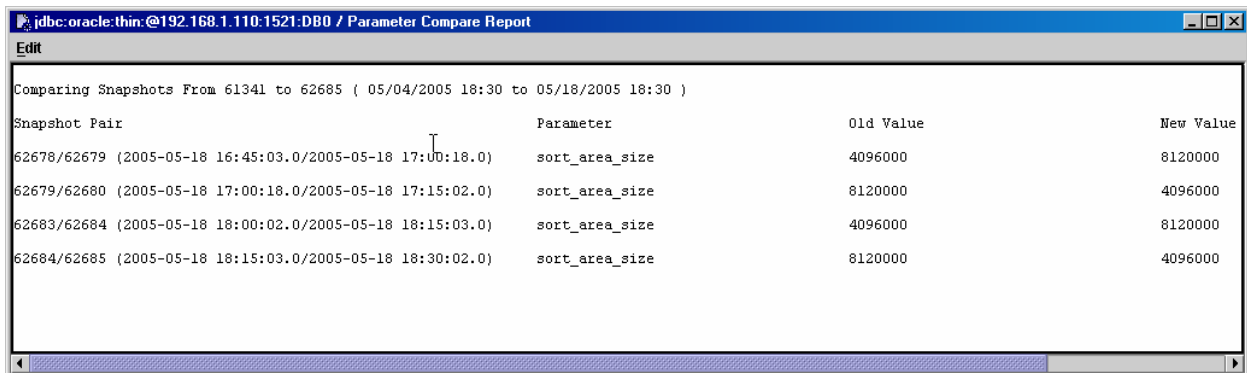
The following screen shows main features of the Parameter Change Analysis panel



The following are the main elements of the Parameter Change Analysis panel:

- A) Generate Parameter Compare report – executes creation of the parameter change analysis report
- B) Timeline Panel (see Timeline panel in “Common Control Elements” chapter)

Below is a sample parameter change analysis report, indicating changes and timeline of changes in the “sort\_area\_size” database parameter



Snapshot Pair	Parameter	Old Value	New Value
62678/62679 (2005-05-18 16:45:03.0/2005-05-18 17:00:18.0)	sort_area_size	4096000	8120000
62679/62680 (2005-05-18 17:00:18.0/2005-05-18 17:15:02.0)	sort_area_size	8120000	4096000
62683/62684 (2005-05-18 18:00:02.0/2005-05-18 18:15:03.0)	sort_area_size	4096000	8120000
62684/62685 (2005-05-18 18:15:03.0/2005-05-18 18:30:02.0)	sort_area_size	8120000	4096000

## Performance Explorer-i SQL Plan/Cost Change Analysis

SQL Plan/Cost Change Analysis enables finding the list of SQL statements that have changed its execution plan or execution cost within a user provided time interval, as well as exact timeline for every change. SQL Plan/Cost Change Analysis helps a user to determine if a change in database performance may be correlated to changes in execution plan or execution cost of certain SQL statements.

### SQL Plan/Cost Change Analysis Panel

The following screen shows main features of the SQL Plan/Cost Change Analysis panel

The following are the main elements of the SQL Plan/Cost Change Analysis panel:

- ”Generate SQL Plan/Cost Change Report”– executes creation of the change report
- Timeline Panel (see Timeline panel in “Common Control Elements” chapter)

Below is a sample SQL Plan/Cost change analysis report, indicating changes and timeline of changes in the cost and hash value of SQL statements

SQL ID	Snapshot Time	Plan Hash Value
8yb3bf20fknmd	03/01/2007 22:50:37	3207614997
8yb3bf20fknmd	03/01/2007 23:50:47	783709846
b07vcvuxryvg9	03/01/2007 22:20:23	3756272070
b07vcvuxryvg9	03/02/2007 00:30:29	2982206256
b07vcvuxryvg9	03/01/2007 22:20:23	3756272070
csdn93bqzjz08	03/01/2007 16:50:11	1940624329
csdn93bqzjz08	03/01/2007 23:50:47	2971472673

SQL ID	Snapshot Time	SQL Cost
0juxuyakquzks	03/01/2007 18:50:09	9
0juxuyakquzks	03/01/2007 23:50:47	8
8yb3bf20fknmd	03/01/2007 22:50:37	27
8yb3bf20fknmd	03/01/2007 23:50:47	24
csdn93bqzjz08	03/01/2007 16:50:11	13
csdn93bqzjz08	03/01/2007 23:50:47	12

## **Performance Explorer-i Visual Root Cause Analysis**

Visual Root Cause Analysis is a standard set of modules that enables a user to see graphic representation of changes in various database metrics. Visual Root Cause Analysis is currently available for:

- System statistics
- Wait Events
- Latches
- Enqueues
- Segment I/O
- Datafile I/O
- Undo Tablespace/ Rollback Segments I/O
- Dictionary Cache
- Library Cache
- Custom user SQL

## Visual Root Cause Analysis Panel

The following screen shows typical visual root cause analysis panels. All panel components are standardized and fully described in “Common Control Elements of Performance Explorer-i for Oracle” chapter of this manual.

- A) Chart Panel
- B) Statistic Selection Panel
- C) Top Search Panel
- D) Timeline/Cross-Time Panels

## Performance Explorer-i ChartBag Reports

Performance Explorer-i ChartBag is a module enabling a user to combine multiple performance metrics from different Performance Explorer-i analysis sections into a single report that can be generated interactively or produced automatically in a batch mode.

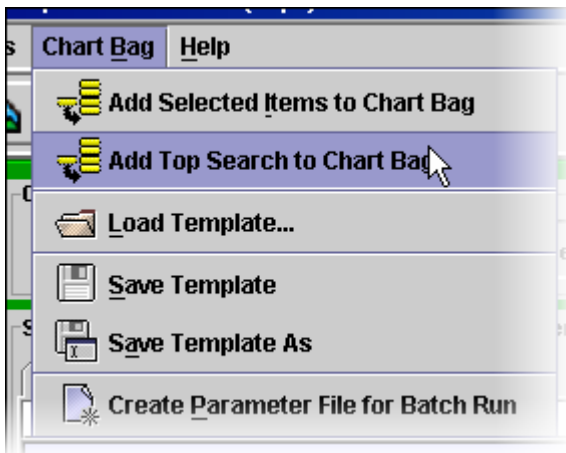
Use of Performance Explorer-i ChartBag is extremely useful for freeing user from routine report generation work, since Performance Explorer-i ChartBag reports can be generated automatically and simultaneously across hundreds of databases, using predefined report templates. Chart Bag module also enables the user to combine data across many databases in a single report.

ChartBag Components can include single database performance metrics, such as system statistics or wait events, SQL statements and SQL statement performance ratios, Top Search results across performance metrics and SQL statements as well as information about database system I/O performance (file systems, tablespaces and data files).

ChartBag Components can be added to the ChartBag Report either by using toolbar button or by selecting menu item **Chart bag->Add selected Item to Chart Bag**. Top search results can be added to ChartBag Report using menu item **Chart bag->Add Top Search to Chart Bag**.



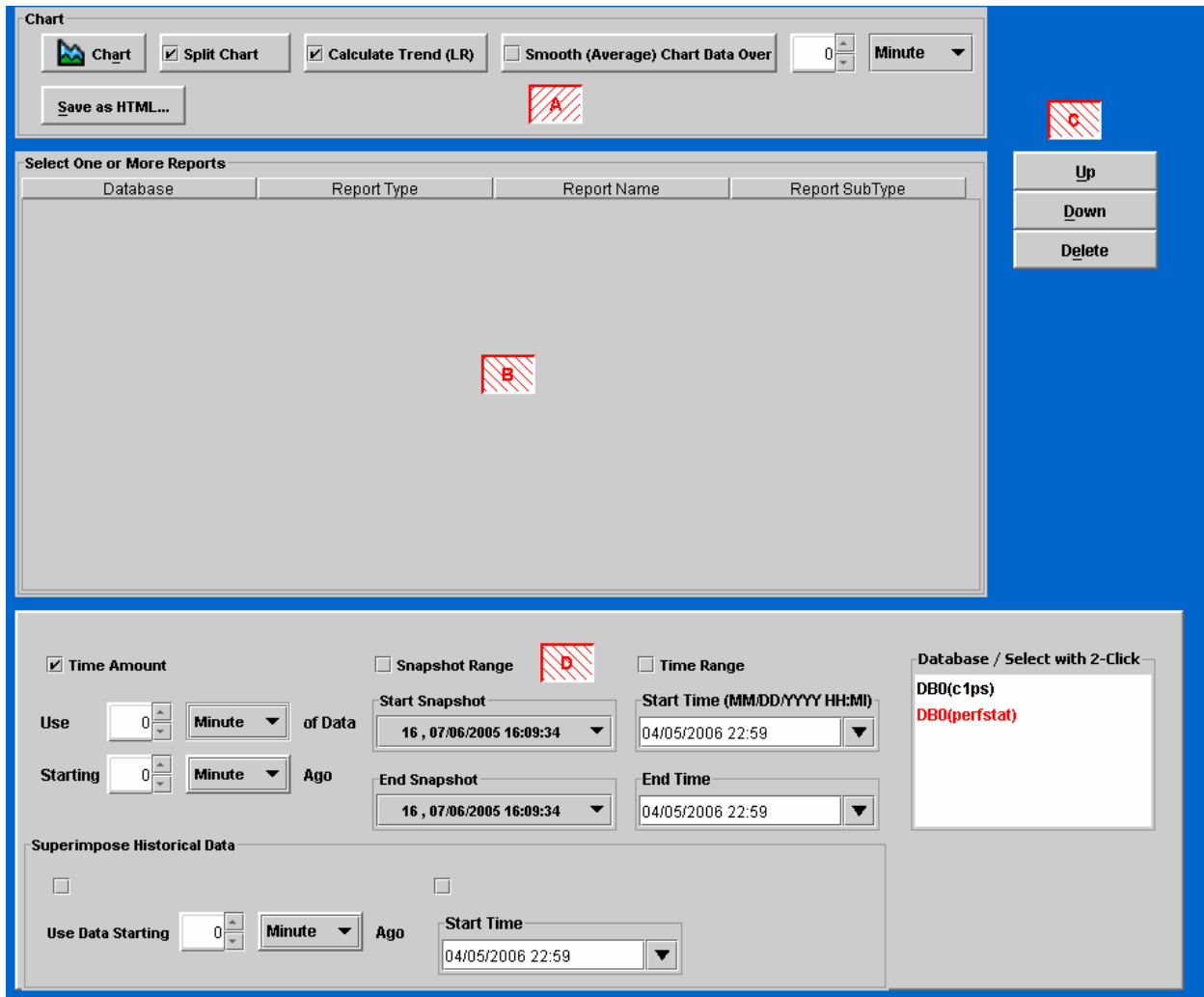
Adding item to ChartBag using toolbar



Adding item to ChartBag using menu selection

## ChartBag Reports Panel

The following screen shows main features of the ChartBag Reports panel

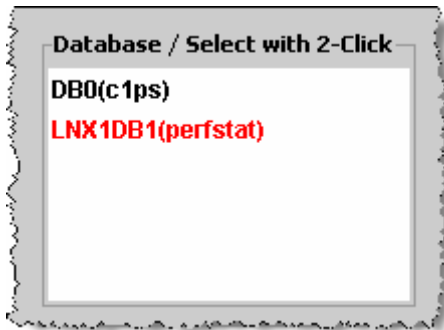


- A) **Chart Control Panel** - described in “Common Control Elements of Performance Explorer-i for Oracle” chapter
- B) **Report Components Table** – contains report components
- C) **Report Components Reordering** block – allows to move report components up/down on the report
- D) **Timeline Control Panel**- described in “Common Control Elements of Performance Explorer-i for Oracle” chapter

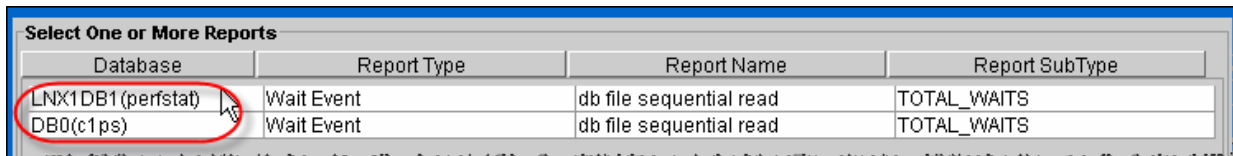
## Performance Explorer-i Cross Database Analysis

Performance Explorer-i allows a user to perform simultaneous analysis of multiple databases. Cross database analysis is supported by snapshot change analysis module and by all visual root cause analysis modules.

Snapshot change analysis supports simultaneous cross-database analysis of two databases – one database to be used as a baseline database and second database to be used as a compared (target) database. Selection of multiple databases for snapshot change analysis is accomplished by selecting databases from the **Database List** of the corresponding timeline sub-panels. The Active database is marked in red.



Visual Root Cause Analysis supports visual cross database analysis through the ChartBag facility. The user can add to the ChartBag multiple performance metrics belonging to different databases. The created ChartBag report displays data from multiple databases.



Database	Report Type	Report Name	Report SubType
LNX1DB1(perfstat)	Wait Event	db file sequential read	TOTAL_WAITS
DB0(c1ps)	Wait Event	db file sequential read	TOTAL_WAITS

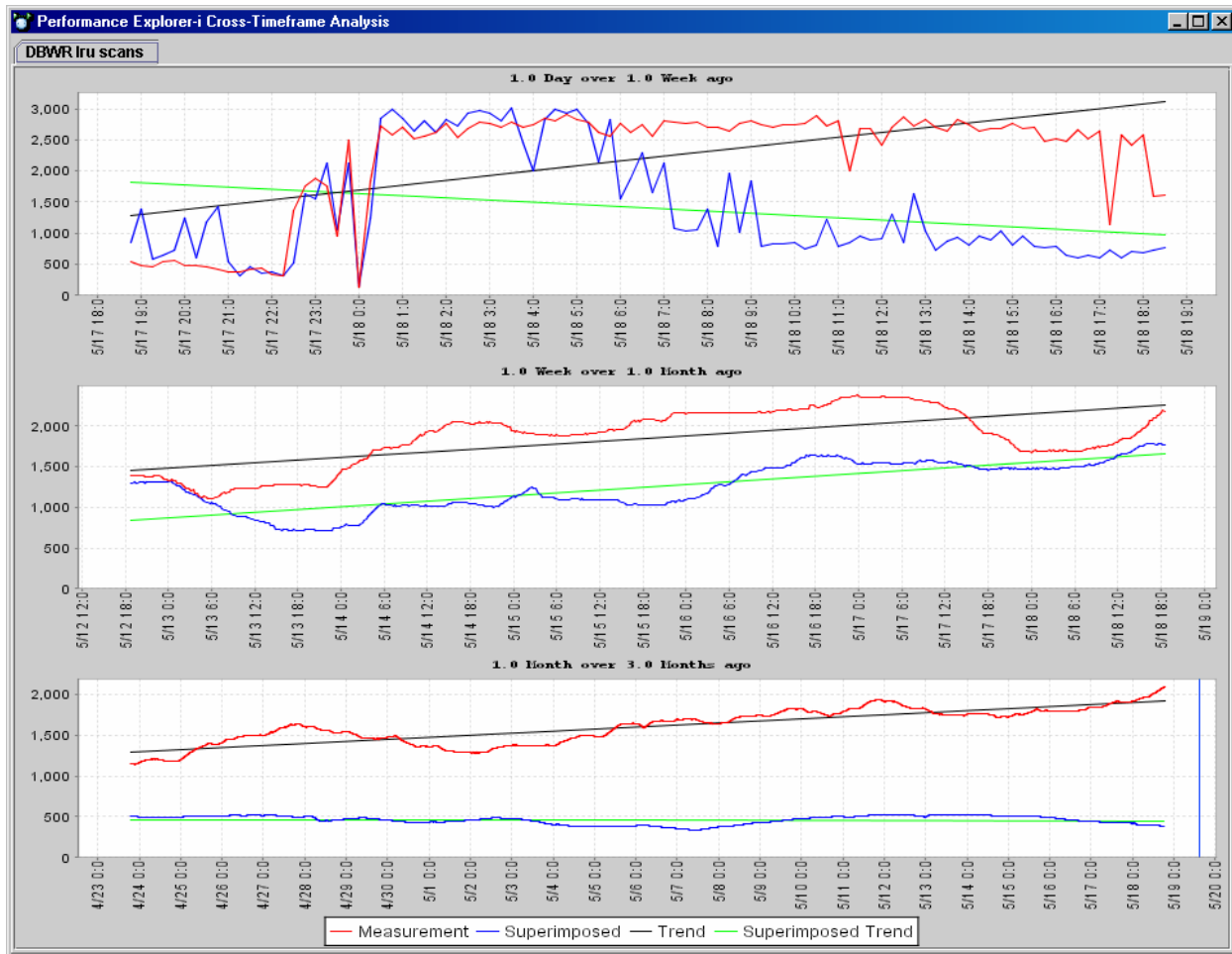
## Performance Explorer-i Cross Time Analysis

When performing Visual Root Cause Analysis, Performance Explorer-i users can perform Cross Time Analysis. Cross Time Analysis allows users to present time periods overlaid with their corresponding historical periods.

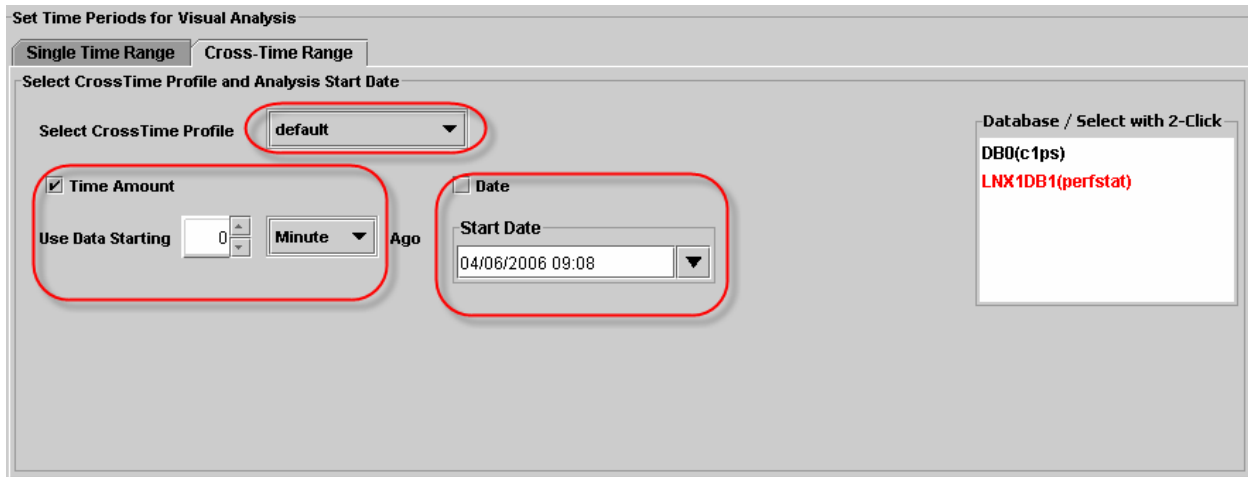
For example, it is possible to create combined analysis graphs using the following cross time ranges:

- 1) Display 1 day of current data over 1 day of data 1 week ago
- 2) Display 1 week of current data over 1 week of data 1 month ago
- 3) Display 1 month of current data over 1 month of data 1 quarter (3 month) ago

Use of Cross Time Analysis is extremely helpful in understanding database and system dynamics, performing headroom and capacity estimates.

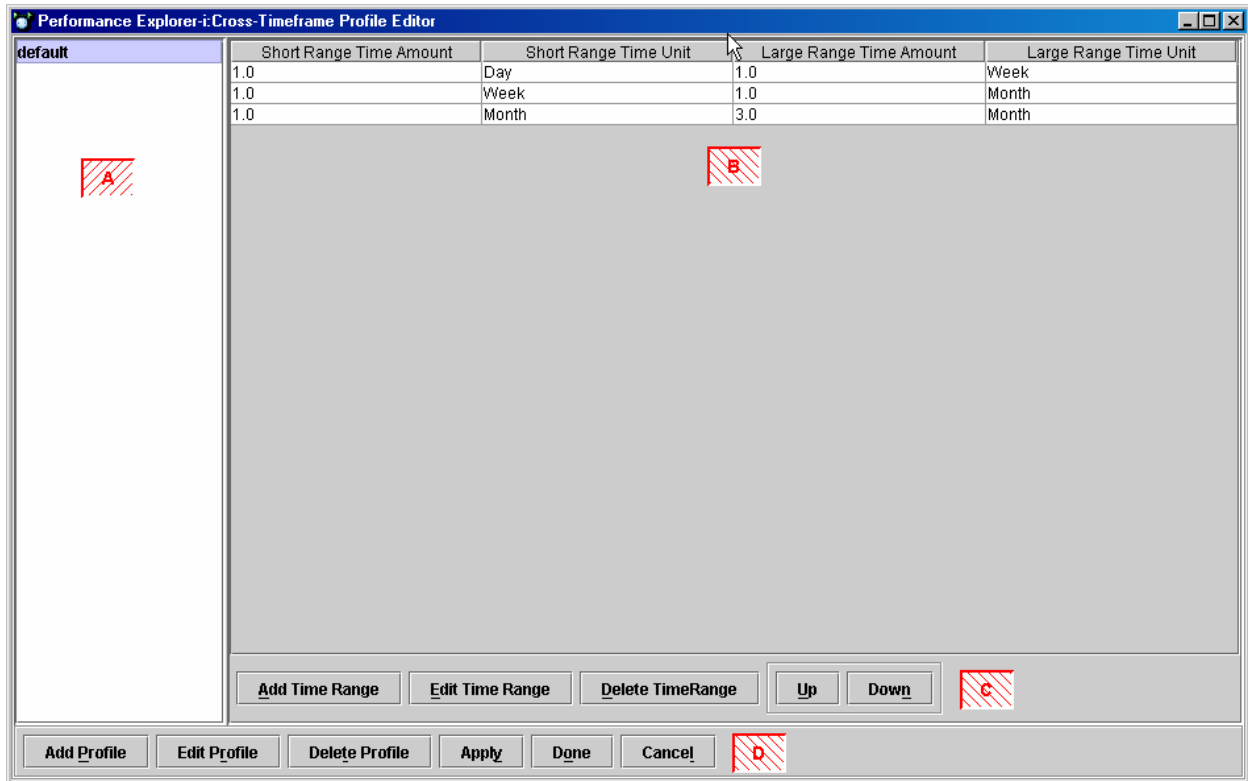


Cross Time Analysis time is executed by selecting Cross-Time Profile option and analysis start time offsets for the cross time analysis on a Cross-Time Range Control sub-panel. (see “Common Control Elements of Performance Explorer-i for Oracle” chapter for panel description). If time amount is set to 0, the analysis is performed on the immediate data.



Cross Time Profile Editor is used to define Cross Time Analysis time ranges. Menu item **Options->Cross Time Profile Editor** is used to open Cross Time Profile Editor.

The following screen shows the main features of the Cross Time Profile Editor:



- A) List of Cross Time profiles
- B) List of Cross Time profile time ranges (per profile)
- C) Time range control block – allows adding, deleting and rearranging time ranges for visual display.
- D) Cross Time profile control block – allows adding, editing, deleting and saving cross time profiles

Cross Time profile time range is added using the dialog below:

The screenshot shows a dialog box titled "Performance Explorer-i:Cross-Timeframe Time Range Editor:Add/Edit Time Range". It contains the following fields and controls:

- Short Time Range Amount:** A numeric input field containing the value "1".
- Short Time Range Unit:** A dropdown menu currently displaying "Day".
- Long Time Range Amount:** A numeric input field containing the value "2".
- Long Time Range Unit:** A dropdown menu currently displaying "Week".

At the bottom of the dialog, there are three buttons: "Apply", "Done", and "Cancel".

Allowed time ranges are:

- a) Minute
- b) Hour
- c) Day
- d) Week
- e) Month

## Using Performance Explorer-i to analyze and troubleshoot your database

Below are sample scenarios demonstrating how Performance Explorer-i can be used to troubleshoot database performance issues and find out the root cause of performance degradation. These scenarios are provided as recommendations for using Performance Explorer-i in production and development environments.

### Database performance degradation

**Case:** At some point in time, database performance and response time become inadequate for business requirements

In such case, we recommend using the following two approaches:

- 1) During immediate problem investigation it is advisable to compare first period of inadequate performance with the closely adjacent period of adequate performance – this investigation helps a user to identify what immediate changes could potentially cause performance and/or availability problems.

For Example; if a database issue started at **14:10 pm** and the performance snapshots are taken every 15 minutes, investigation could use time period of **13:45 to 14:00 pm** as a **baseline period** and time period of **14:00 to 14:15 pm** as a compare period.

- 2) Considering that database load is cyclical in nature, corresponding to the business cycles (as on many OLTP systems) it is recommended to compare problematic period of time with the comparable period of time when the database performance was adequate. If database operates in a **weekly cycle**, it is possible to compare performance of current period of time with the similar period of time one week ago. One day or one month, etc period could also be used.

For Example; if database issues started on 04/05/2006 at 15:35 pm, and the performance snapshot are taken every 15 minutes, an investigation could be done by using time periods of **03/30/2006 15:30 to 15:00 pm** as a **baseline period** and time period of 04/05/2006 **15:30 to 15:45 pm** as a compared period.

## **Database performance degradation within database cluster/logical group**

**Case:** At some point in time, database performance and response time become inadequate for business requirements, while performance of other nodes in database cluster/group remains acceptable.

In such case, we recommend to use the following two approaches:

- 1) Using option of cross-database performance analysis – use adequately performing database cluster node as a baseline mode and problematic node as a compare node:

For Example; if database node **DB0N1** started having problems at 15:35 pm, and performance snapshots are taken every 15 minutes, investigation could use time periods of **15:30 to 15:00 pm on node DB0N0** as a **baseline period** and time period of **15:30 to 15:45 pm on node DB0N1** as a compared period.

- 2) Using approach described above in the “Single database performance degradation” to identify root cause of the changes comparing to the time period for the same database when performance was acceptable,

## **Database performance degradation after database upgrade**

**Case:** After a database is upgraded, database performance becomes inadequate.

In such case, we recommend to:

Compare database performance related to time period before the upgrade (baseline) to the database performance after the upgrade.

## **Database performance degradation within logical group of databases, when database versions are different**

**Case:** After production database upgrade, performance of database in logical group could become inadequate for business requirements comparing to different nodes in the database group.

In this case, we recommend to do the following:

Utilize Cross Database Analysis option – use adequately performing database in the group as a baseline and problematic database as a compared node.

## **Database performance degradation after application upgrade**

**Case:** After application accessing the database is upgraded, database performance becomes inadequate.

In such case, we recommend to:

Compare database performance related to the time period before the upgrade (baseline) to the database performance after the upgrade.

## **QA database performance/stress testing**

**Case:** Application developer needs to understand how new application code is loading database comparing to the old application code.

Our recommendation:

Compare database performance in the time period during the old application code run (baseline) to the database performance related to run of the new application code.

## **QA database performance/stress testing during database upgrade**

**Case:** User needs to understand how new application code is loading upgraded database comparing to the pre-upgrade time.

Our recommendation:

Compare database performance during a pre-upgrade time period (baseline) to the database performance in the post-upgrade period.

## **Using Snapshot Compare Report data for Visual Root Cause Analysis**

Change root cause analysis report provides the user with detailed change information for the selected areas of database performance. During the root cause investigation, it is extremely important to review areas of analysis that show significant changes in performance metrics values, such as changes in SQL statistics, wait events, etc. Another important area to check is the “NEW” report sections that contain all new performance metrics that were not present in the database during the baseline time period.

## Finding Top Resource Consuming SQL

Top SQL search allows a user to perform very granular filtering of top SQL by specifying a search pattern and requesting the search to include only DML statements. This flexibility enables the user to spot with great precision which SQL has most impact on database performance and stability.

In the below example the user's request is to perform top executions search on DML statements that include "ITEM\_ATTR" pattern (table name in the given case).

**Search Panel**

SQL Search Pattern (used by Regular and TOP SQL Search)

DML Search Only

HASH Value Search (single value or comma separated list below)

ITEM\_ATTR

Search Memory

Search Statspack

Create Custom SQL Statistics Ratio

**Top Search**

Top Limit: 20

In Window

**SQL Statistics**

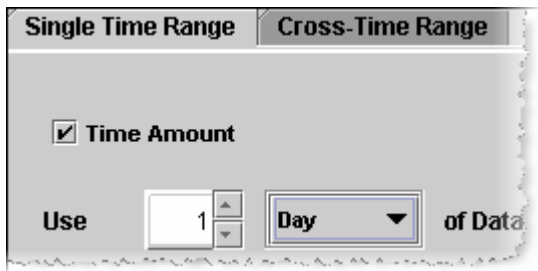
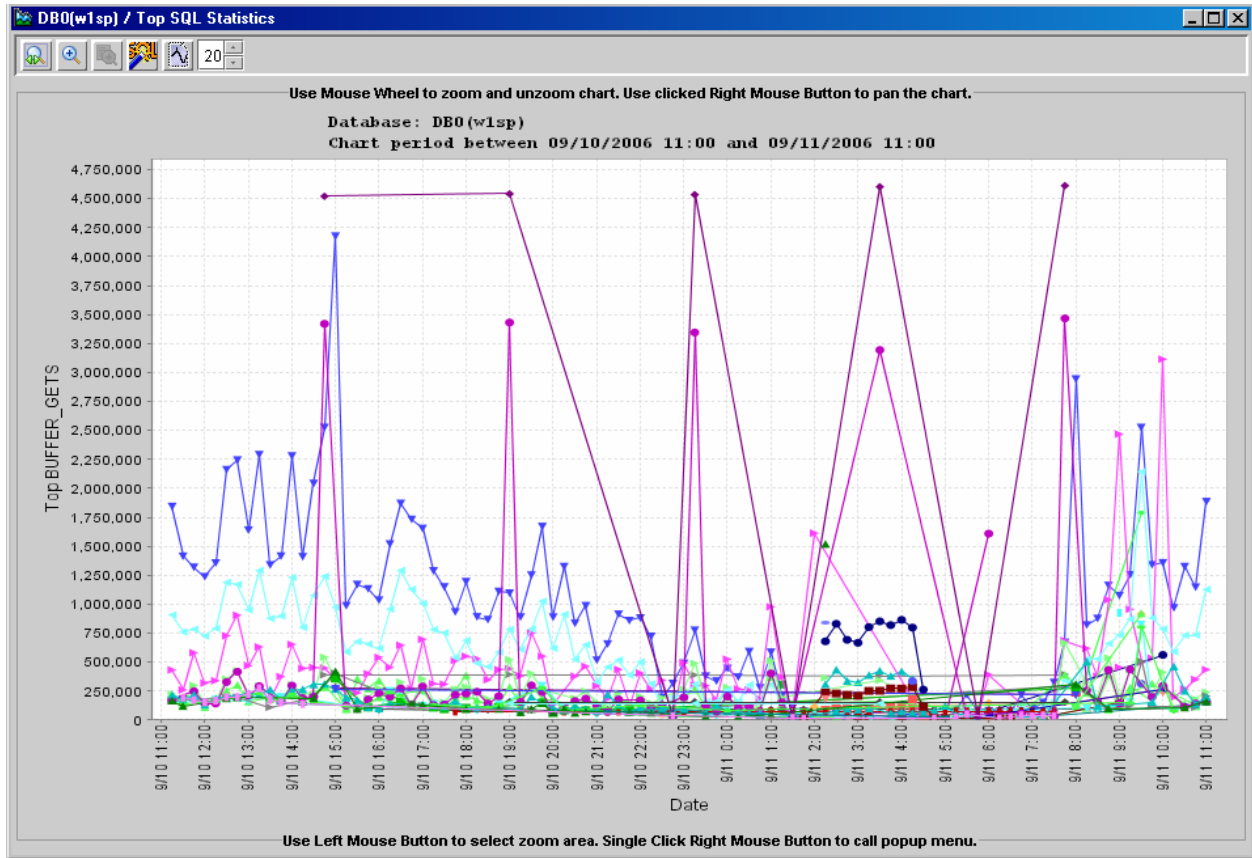
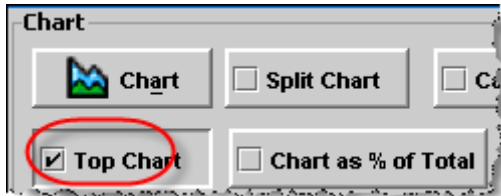
- EXECUTIONS
- BUFFER\_GETS
- DISK\_READS
- ROWS\_PROCESSED
- SORTS
- PARSE\_CALLS
- LOADED\_VERSIONS
- LOADS
- INVALIDATIONS
- VERSION\_COUNT
- SHARABLE\_MEM
- BUFFER\_GETS/EXECUTIONS
- DISK\_READS/EXECUTIONS
- SORTS/EXECUTIONS

**SQL (Ctrl-C to Copy) / Select One or More Statements to Chart or Explain**

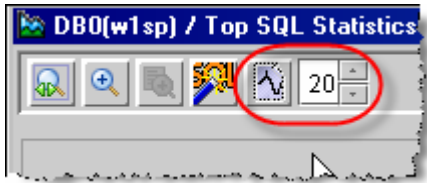
Hash Value	SQL (Hold Mouse Over to see the whole statement)	Top V...	Total...	Avg. B...	Avg. Di...
1138735236	INSERT INTO /* SaleAggregatorMap.INSERTPK_WIT...	2,449	2,449	8.71	0.02
2307645813	UPDATE /* SaleAggregatorMap.UPDATEPK_WITH_B...	446	446	7.76	0
2829196054	UPDATE /* SaleAggregatorMap.UPDATEPK_WITH_B...	160	160	7.8	0
2878067757	INSERT INTO /* SaleAggregatorMap.INSERTPK_WIT...	32	32	7.95	0.02
4068681277	INSERT INTO /* SaleAggregatorMap.INSERTPK-2 [e...	27	27	6.18	0.59
3150052513	UPDATE /* SaleAggregatorMap.UPDATEPK-2 [e Da...	12	12	7.82	0
3864397275	DELETE /* SaleAggregatorMap.DELETEPK-1 [e Dan...	0	0	10.35	0.34

## Performing Time Range Top SQL Analysis

Top SQL Analysis allows user to perform very granular identification of Top resource consuming SQL statement for every time point of the analysis time range



Top SQL Analysis can be combined with Top SQL Spike report that allows user to identify exact SQL statement and SQL execution plans in the selected time range



**Top SQL Spike Report**

**20 Top BUFFER\_GETS Report**  
Time period between  
**09/11/2006 01:13:57 and 09/11/2006 05:17:27**

2735175695	09/11/2006 02:00:00	1,614,237
------------	---------------------	-----------

```

declare be cris_be_type := int_upload.init_status_obj(); begin be.resort := :in_be_resort; be.modulename := :in_be_module;
be.action_type := :in_be_action; be.action_instance_id := :in_be_action_id; be.created_date := to_date(:in_be_created_date,
'YYYYMMDDHH24MISS'); be.created_time := to_date(:in_be_created_time, 'YYYYMMDDHH24MISS'); be.primary_key := :in_be_primary_key;
be.parent_action_instance_id := :in_be_parent_action_id; :out_ret := int_xml_m\n.process_outbound(:in_interface_id, be,
:out_msg_generated, :out_stop_requested, :out_err_msg); exception when others then :out_err_msg := sqlerrm(); :out_ret := -99;
end;
    
```

SQL Plan N/A

2589943145	09/11/2006 02:15:01	1,520,309
------------	---------------------	-----------

```

DECLARE job BINARY_INTEGER := :job; next_date DATE := :mydate; broken BOOLEAN := FALSE; BEGIN int_purge.AutoPurge; :mydate :=
next_date; IF broken THEN :b := 1; ELSE :b := 0; END IF; END;
    
```

SQL Plan N/A

505428634	09/11/2006 04:00:00	865,398
-----------	---------------------	---------

```

BEGIN :outResult := Business_Event.Ext_Get_Event_XML(:outEvent, :inWaitMode, :pull); END;
    
```

## Finding Time Period Top System Statistics

Select One or More Statistics / Type Name to Filter

**System Statistics** | **Top System Statistics**

Name	Top Value
process last non-idle time	1,304,169,020,424
bytes sent via SQL*Net to client	3,412,877,903
table scan rows gotten	683,173,978
bytes received via SQL*Net from client	247,131,358
redo size	184,484,080
consistent gets	57,151,577
session logical reads	56,382,817
no work - consistent read gets	44,532,808
buffer is not pinned count	35,501,095
buffer is pinned count	33,144,174

**Top Search**

Top Limit:

In Window

**Set Time Periods for Visual Analysis**

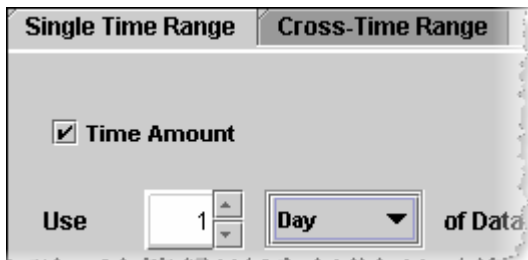
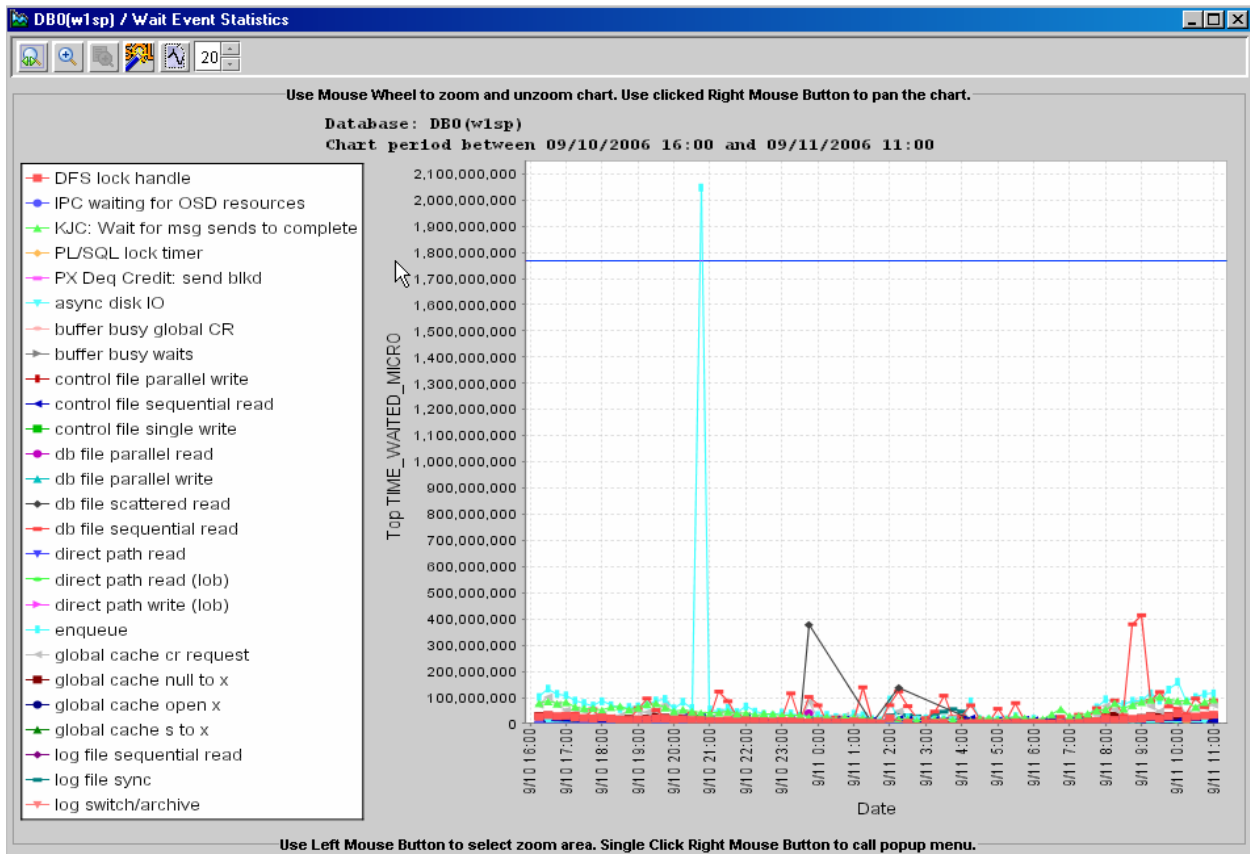
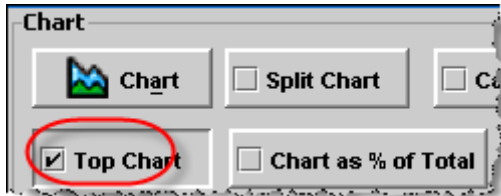
**Single Time Range** | **Cross-Time Range**

Time Amount

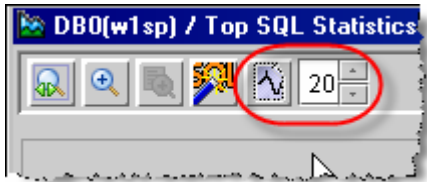
Use    **Hour** of Data

## Performing Time Range Top Wait Event Analysis

Top Wait Event Analysis allows user to perform very granular identification of Top resource consuming wait events for every time point of the analysis time range



Top Wait Event analysis can be combined with Top Wait Analysis report that allows user to identify exact top waits events in the selected time range



**Top Wait Event Spike Report**

**20 Top TIME\_WAITED\_MICRO Report**  
**Time period between**  
**09/11/2006 06:05:30 and 09/11/2006 10:21:48**

Metric	Date/Time	Value
db file sequential read	09/11/2006 09:00:01	414,391,189
enqueue	09/11/2006 10:00:00	159,666,894
KJC: Wait for msg sends to complete	09/11/2006 09:30:01	102,707,293
global cache cr request	09/11/2006 09:00:01	87,899,085
DFS lock handle	09/11/2006 10:00:00	39,880,032
global cache null to x	09/11/2006 08:15:01	30,654,318
control file sequential read	09/11/2006 09:15:00	20,217,730
log file sync	09/11/2006 09:45:00	11,663,994
log file sequential read	09/11/2006 09:45:00	8,434,535
global cache open x	09/11/2006 09:00:01	7,859,513
direct path read (lob)	09/11/2006 10:00:00	7,263,273
PL/SQL lock timer	09/11/2006 10:15:02	5,984,344
async disk IO	09/11/2006 08:45:01	5,773,302
global cache s to x	09/11/2006 10:15:02	5,550,798
IPC waiting for OSD resources	09/11/2006 08:15:01	3,974,625
db file parallel write	09/11/2006 09:30:01	2,324,933
db file scattered read	09/11/2006 08:30:00	1,596,746
control file parallel write	09/11/2006 06:30:00	1,346,635
direct path read	09/11/2006 06:30:00	499,378
PX Deq Credit: send blkd	09/11/2006 07:00:00	469,494

# Finding Time Period Top Wait Events

**Set Time Periods for Visual Analysis**

Single Time Range    Cross-Time Range

Time Amount

Use  Hour of Data

Select One or More Statistics / Type Name to Filter

Wait Events    Top Wait Events

Name	Top Value	% of Top Total
SQL*Net message to client	1,269,119	24.77
SQL*Net message from client	1,269,081	24.77
db file sequential read	1,101,744	21.51
SQL*Net more data to client	1,093,767	21.35
rdbms ipc message	146,310	2.86
file open	89,486	1.75
log file sync	64,601	1.27
log file parallel write	60,663	1.19
SQL*Net more data from client	23,054	0.45
log file sequential read	6,404	0.13

Wait Events Statistics

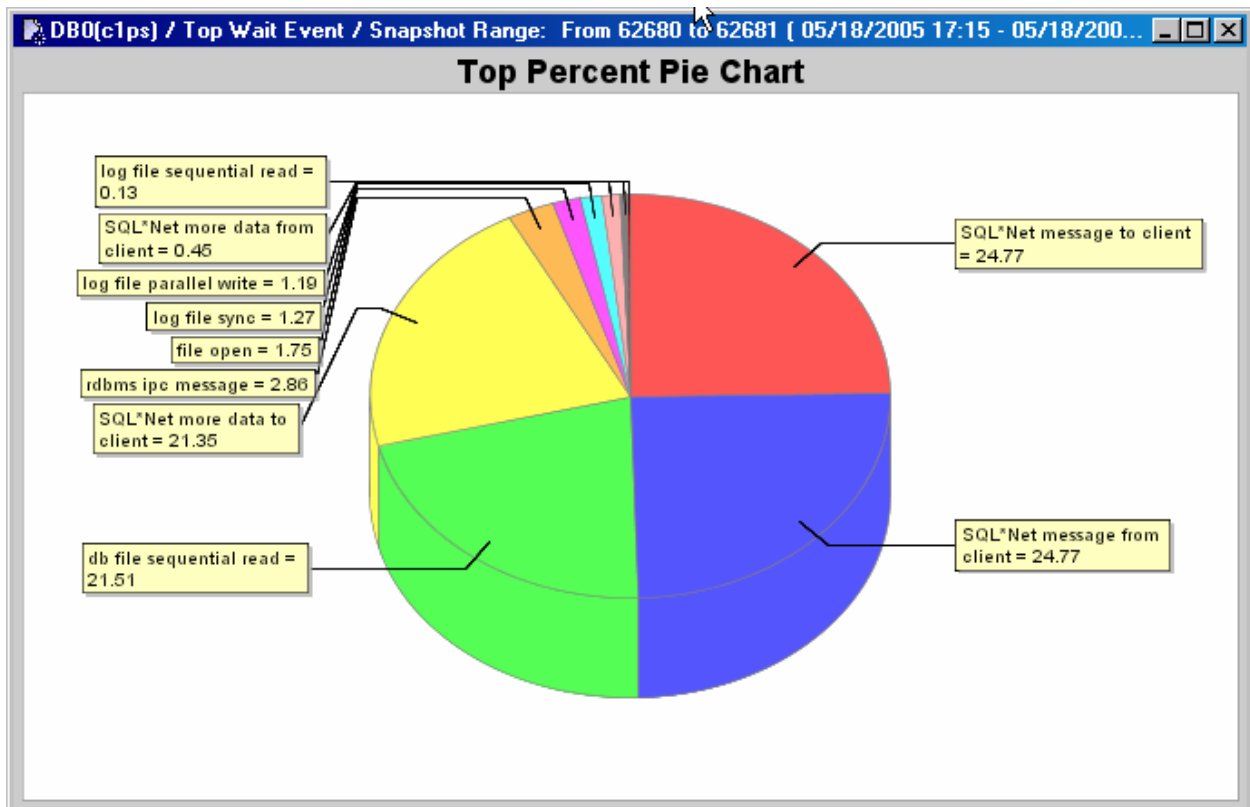
- TOTAL\_WAITS
- TOTAL\_TIMEOUTS
- TIME\_WAITED
- AVERAGE\_WAIT\_TIME
- AVERAGE\_TIMEOUT\_TIME

Top Search

Top Limit: 10

In Window

Pie Chart



# Finding Time period Top Latches

**Set Time Periods for Visual Analysis**

Single Time Range    Cross-Time Range

Time Amount

Use  Hour of Data

Select One or More Statistics / Type Name to Filter

**Latches**    **Top Latches**

Name	Top V...	% of To...
library cache	251	91.95
cache buffers chains	11	4.03
shared pool	9	3.3
session idle bit	1	0.37
transaction allocation	1	0.37
AQ Propagation Scheduling Proc Table	0	0
AQ Propagation Scheduling System Load	0	0
Direct I/O Adaptor	0	0
GDS latch	0	0
JOX SGA heap latch	0	0

**Latche Statistics**

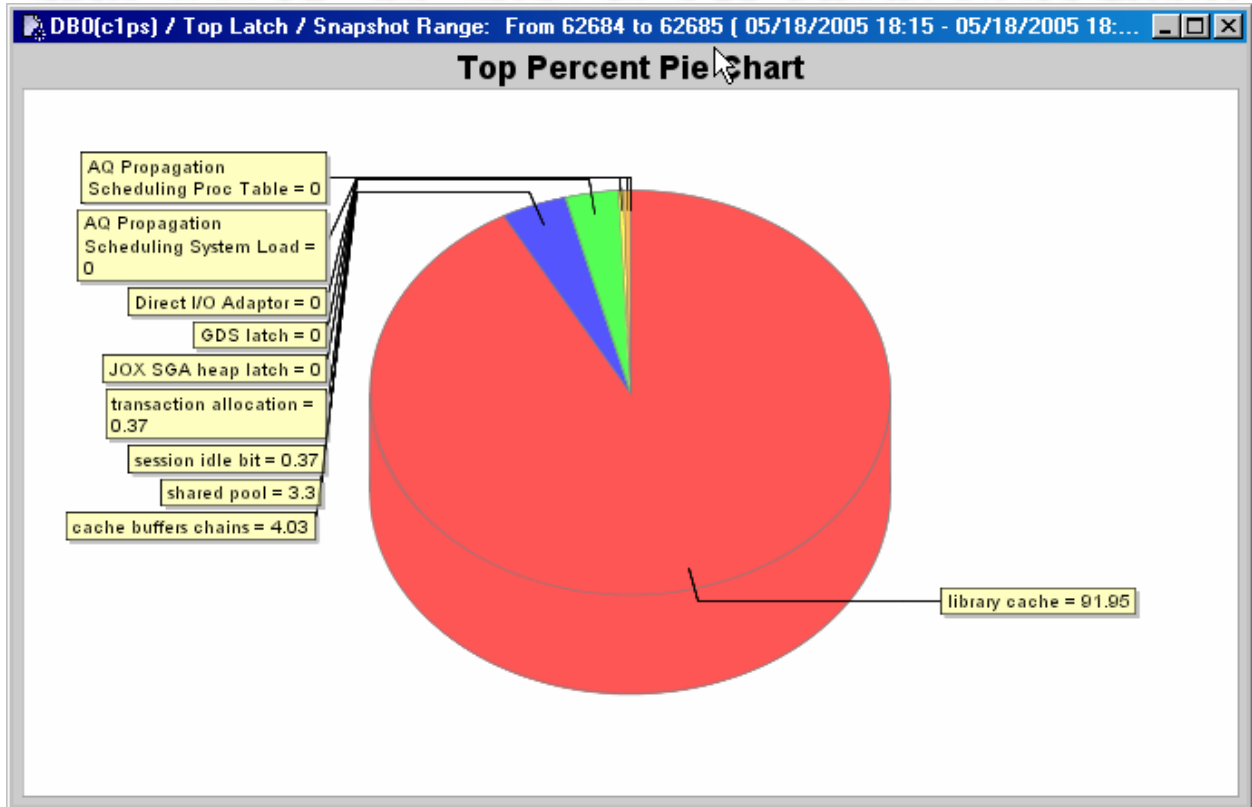
- GETS
- MISSES
- SLEEPS
- IMMEDIATE\_GETS
- IMMEDIATE\_MISSES
- SPIN\_GETS
- SLEEP1
- SLEEP2
- SLEEP3
- SLEEP4
- GETS/MISSES
- GETS/SLEEPS
- IMMEDIATE\_GETS/IMMEDIATE\_...
- MISSES/GETS
- SLEEPS/GETS

**Top Search**

Top Limit:

In Window

Pie Chart

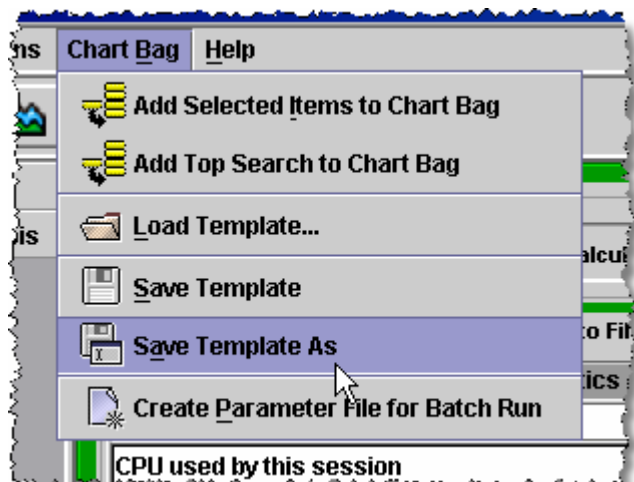


## Using Performance Explorer-i for performance reporting

ChartBag module is used for standardized performance reporting. After a set of reporting parameters is identified, it could be saved in ChartBag Template and reused across multiple databases.

Menu items **ChartBag->Save Template** and **ChartBag->Save Template As...** can be used to create a chart bag template.

Menu item **ChartBag->Load Template** can be used to load an existing template.



## Using Performance Explorer-i for the creation of performance and headroom web site

ChartBag reports can be executed in a batch mode across multiple databases. The resulting report is generated as a set of HTML files that could be published immediately on a user web site. To execute ChartBag reports in a batch mode, ChartBag Parameter File needs to be in place. **ChartBag parameter file** could be created by using menu item **ChartBag->Create Parameter File for Batch Run** or by creating a text file with necessary parameters

### ChartBag Parameter file supported parameters are:

#	Comment
Db	Database connection URL
User	Database user name
Password	Database user password
Output	Base name for the set of output files
Width	Width of the created graph images in pixels
Height	Height of the created graph images in pixels

Sample Batch parameter file:

```
#>>
# DB1 headroom collection parameter file
#
db=jdbc:oracle:thin:@192.168.1.200:1521:LNx2DB1
user=perfstat
password=perfstat
chartbag=DB1_Headroom.cbg
output=batch_hdr_cbg_db1
width=1024
height=1024
#<<
```

Performance Explorer-i automatic report generation can be executed in a batch mode using:

A) Using crontab (UNIX) automatic report generation as:

**pei<v>.sh parfile=<parameter\_file\_name> , where v is Oracle Client version**

For example:

**pei9.sh parfile=weekly\_db0.cbg**

B) Using windows task scheduler schedule as:

**pei<v>.bat parfile=<parameter\_file\_name> , where v is Oracle Client version**

For example:

**pei9.bat parfile=weekly\_db0.cbg**

Output HTML files are generated in the "html" directory under Performance Explorer-i installation directory.

## Using Performance Explorer-i in secure environment (SSH tunneling)

When direct TNS connection from desktop to database is not allowed due to security reasons, a user can create SSH tunnel in order to connect to a database on a remote host. SSH Tunnel can be created using "putty", "Secure CRT" or any other SSH terminal software that supports tunnel creation.

For example, to set up SSH tunnel for default database listener port 1521:

In Putty: select "**SSH->Tunnels**" Category, set "**Source port**" to **1521**" and Destination to "**db\_host:1521**".

In "Secure CRT": select "**Session Options**", "**Advanced**" , "**Port Forwarding**" tab, set "**Local Port**" to **1521**, remote hostname to "**db\_host**" and "**Remote port**" to **1521** as well.

After SSH tunnel is set up, tnsnames.ora entry needs to be changed to have "127.0.0.1" or "localhost" instead of <db\_host> name. For thin URL connections, please use "127.0.0.1" or "localhost" as well.



# 6

## Troubleshooting Performance Explorer-i for Oracle

This chapter contains information that can assist the user with issues that are related to Performance Explorer-i startup and its connection to the databases.

<b>In this chapter</b>	<b>Page</b>
Installation Issues	62
Startup Issues	62
Connectivity Issues	62
Other Known Issues	62

## **Installation related issues**

If during the installation process Oracle Enterprise Manager client is active, the installer may stall. To resolve this issue, close Oracle EM client during the installation process

## **Startup related issues**

Performance Explorer-i may fail to startup if during the installation process the user chooses not to install the provided JRE (Java Runtime Environment). It is strongly recommended to use JRE packaged with Performance Explorer-i.

## **Connectivity related issues**

Sometimes, due to minor differences between the provided JDBC drivers and the installed Oracle client, connection to database, using tnsnames connection name cannot be successfully established.

In this case, it is recommended to copy JDBC drivers from Oracle client installation to the "Performance Explorer\lib" and rename them to the proper classes12<oracle version>.jar name, i.e. classes129.jar is a proper name for Oracle 9i and classes1210.jar is a proper name for Oracle 10g

## **Other known issues**

Negative value for SQL statistics displayed as a result of the Top SQL Search

The SQL Analysis Top SQL Search component of the Performance Explorer-i may display a negative value on a SQL Top Search Statistic or on a displayed SQL ratio. This occurs when in a heavily used database system the value of the SQL performance statistic counter is growing over  $4 \times 10^9$  and becomes negative. This affects all Oracle versions 8i, 9i, and 10g.

