



Embarcadero Performance Center 2.7.2 User Guide

Copyright © 1994- Embarcadero Technologies, Inc.

Embarcadero Technologies, Inc.
100 California Street, 12th Floor
San Francisco, CA 94111 U.S.A.
All rights reserved.

All brands and product names are trademarks or registered trademarks of their respective owners. This software/documentation contains proprietary information of Embarcadero Technologies, Inc.; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with Restricted Rights, as defined in FAR 552.227-14, Rights in Data-General, including Alternate III (June 1987).

Information in this document is subject to change without notice. Revisions may be issued to advise of such changes and additions. Embarcadero Technologies, Inc. does not warrant that this documentation is error-free.

Contents

- Welcome to Performance Center 22
 - About This Document 22
 - Product Benefits by Audience 22
 - Additional Product Information 22
 - About This Guide 23
- Performance Center Overview 26
 - Application Design 26
 - Application Interface 27
 - Views 29
 - Workbook 51
 - Output Window 51
 - Status Bar 53
 - Event Log 53
 - Explorer 54
 - Product Navigation 56
 - Menu Bar 56
 - Shortcut Menus 72
 - Toolbar 72
 - Maintenance Dialog 72
 - Address Maintenance 73
 - Notification Maintenance 73
 - Action Maintenance 73
 - Statistic Maintenance 75
 - Schedule Maintenance 75
 - Report Maintenance 76
 - Data Extraction Jobs 76
 - Alert Template Maintenance 76
 - Health Index Template Maintenance 84
 - Recorded Statistic Template Maintenance 90
 - Configuring Performance Center Client 90
 - General Options 91
 - Statistic Options 92
 - Category Options 93
 - Datasource Options 93
 - Editor Options 93
 - Monitor Server Options 94

Configuring Performance Center Server	94
Run As Options	95
Analyzer Options	96
Capture Agent Options	96
Repository Options	97
Datasource Manager Options	98
Emergency Contact Options	98
Notifier Options	99
Requestor Options	102
Report Options	102
Using Performance Center	104
Datasource Management	104
Selecting Datasources	105
Managing Datasources	105
Working with Datasources	120
Organizing Datasources	121
Statistics	125
Opening the Statistic Maintenance dialog	125
Completing the Statistic Maintenance dialog	126
Statistic Properties dialog	126
Opening the Statistic Properties dialog	128
Health Indexes	128
Health Index Properties	128
Health Index Templates	130
Addresses	135
Opening the Address Maintenance dialog	136
Adding or Editing an Address	136
Attaching an Address to a Notification	139
Deleting an Address	140
Notifications	141
Opening the Notification Maintenance dialog	142
Adding or Editing a Notification	142
Tivoli Notifications	143
Attaching a Notification	147
Deleting a Notification	147
Actions	148
Using the Action Scripts Setup	149
Opening the Action Maintenance dialog	163
Adding or Editing an Action	163

Deleting an Action	165
Schedules	165
Opening the Schedule Maintenance dialog	166
Adding or Editing a Blackout Schedule	167
Attaching a Blackout Schedule	169
Detaching a Blackout Schedule	171
Deleting a Blackout Schedule	171
SQL Capture	172
Setting Up the Session Length	172
Setting Up the Capture Interval	174
Setting a SQL Capture Session on an Individual Alert:	174
Gaining Access to the Captured Information	176
Enabling SQL Capture for Oracle/Sybase/DB2/Microsoft SQL Server	178
Datasource Alerts	180
Opening the Datasource Alerts Maintenance dialog	181
Adding or Editing an Alert	182
Deleting a Datasource Alert	183
Alert Templates	184
Recorded Statistic Templates	185
Recorded Statistic Template Maintenance dialog	185
Datasource Recorded Statistics dialog	186
Adding or Editing an Alert or Recorded Statistic Template	187
Deleting a Alert Template or Recorded Statistic Template	190
Cloning an Alert Template or Recorded Statistic Template	191
Applying an Alert, Health Index Template, or Recorded Statistic Template	191
Reports	192
Using the Report Maintenance dialog	193
Select Historical Report Wizard	193
Logging into the Oracle Statspack	195
Data Extraction Jobs	195
Data Extraction Jobs dialog	195
Archive	201
Alarms	201
Alarms Criteria dialog	201
Acknowledging Alarms	203
Alert Log/SQL Server Logs/Sybase Error Log/DB2 Diagnostic Log	204
Completing the Find dialog in the Alert Log/SQL Server Logs/Sybase Error Log View	206
Completing the Alert Log Information dialog	206
Completing the Sybase Error Log Information dialog	207

Health Index	207
Instance Parameters/Server Configuration/Configuration	207
Locks.....	207
Object Extents Tab	208
Top Sessions.....	208
Session Detail.....	208
Top SQL	209
Customizing the Top SQL Display	210
Completing the Find dialog in the Top SQL View	211
Running an Explain Plan/Query Plan on SQL Statements	211
Profiled SQL	211
Trends.....	212
Completing the Add Statistic to Chart dialog	213
Deleting a Live Statistic from the Trends Chart.....	213
SQL Scripting	214
SQL Editor	214
Results Editor	223
Performance Center Web Client.....	231
Enterprise View.....	231
Performance Center Web Client Views	232
Monitor Details	232
Server Details	233
Client Details.....	234
Admin Login	234
Reports	234
Help.....	235
Microsoft SQL Server Expert Guide	236
Home View Statistics	236
Memory Vital Signs.....	237
I/O Vital Signs	239
Contention Vital Signs.....	240
Space Vital Signs	242
Users Vital Signs.....	243
Network Vital Signs.....	245
Memory Page Statistics	246
Buffer Cache Hit Ratio	246
Procedure Plan Hit Ratio.....	247
Ad Hoc SQL Hit Ratio	247

SQL Cache	248
Memory Usage	248
Buffer Cache	249
Session Leaders - Memory	249
Cache Usage	250
SQL Cache Size	250
SQL Cache Free	250
Memory Detail View	251
I/O Page Statistics	253
Page Reads	254
Page Writes	255
Read Ahead Pages	255
Log Flushes	256
Log Cache Reads	256
Checkpoint Pages	256
Full Scans	257
Range Scans	258
Index Searches	258
Probe Scans	258
Worktables Created	258
Forwarded Record Fetches	259
Server I/O Busy Rate	259
Session Leaders - I/O	259
Total I/O	260
I/O Error Rate	260
I/O Detail View	260
Database I/O Tab	261
Space Page Statistics	264
Disk Free Space	265
Database Overview	265
Log Overview	266
Space Detail View	267
File Groups/Files Tab	267
Users Page Statistics	271
Total Connections	272
Active Transactions	273
T-SQLBatches	273
Active Connections	273
Inactive Connections	273
System Connections	273

SQL Compilations	274
SQL Re-Compilations	274
Auto-Param Attempts	275
Failed Auto-Param Attempts	275
Users Blocked	276
Session Leaders - Memory	276
Session Leaders - I/O	277
Session Leaders - CPU	277
Users Detail View	277
Contention Page Statistics	280
Blocked Users	281
Page Deadlocks	281
Number of Deadlocks	282
Latch Waits	282
Log Flush Waits	283
Table Lock Escalations	283
Lock Overview	283
CPU Busy	284
I/O Busy	284
Server Idle	284
Contention Detail View	284
Database Page Statistics	288
Errors in Current Error Log	289
Databases	289
Files	289
File Groups	290
Suspect Databases	290
Offline Databases	291
Databases Needing Backup	291
Databases Without Auto-Growth	291
Logs Without Auto-Growth	292
Databases Without Auto-Update Statistics	292
Databases Without Auto-Creat Stats	293
Database Summary	293
Database Detail View	294
Network Statistics	299
Packets Received	300
Packets Sent	300
Network Reads	300
Network Writes	300

Network Bytes Read	301
Network Bytes Written	301
Packet Errors	301
Total Traffic	301
Network I/O Wait Requests	301
Network I/O Wait Time	302
OS Page Statistics	302
Summary Tab	302
Processor Time	303
Processor Speed	303
Processor	304
Disk Time	304
Load Average	304
Paged Memory Used	304
Number of Processors	304
Swap Memory Used	304
Average Disk Queue	305
Page Faults/Sec	305
Processor Queue	305
Network Output Queue/Network Queue	306
Available Physical Memory	306
Available Paged Memory	306
Available Swap Memory	307
Total Physical Memory	307
Total Paged Memory/Total Swap Memory	307
Used Disk Space	307
Total Disk Space	308
Free Disk Space	308
Processes Overview	308
Number of Logins	309
Number of Processes	309
Sybase Expert Guide	318
Home View Statistics - Sybase	318
Memory Vital Signs	318
Contention Vital Signs	321
I/O Vital Signs	324
Users Vital Signs	325
Space Vital Signs	326
Network Vital Signs	327

Memory Statistics - Sybase	328
Data Cache Activity	329
New Pages Allocated	330
Dirty Buffers Grabbed	330
Dirty Read Requests	330
Dirty Read Restarts	331
ProcedureCache Activity	331
Procedure Requests	332
Procedure Reads from Disk	332
Procedure Writes to Disk	333
Procedure Removals	333
Large I/O Hit Rate	333
Session Leaders - Memory	334
Cache Allocations	335
Memory Detail View	335
I/O Statistics - Sybase	338
Requested Disk I/Os	338
Completed Disk I/Os	338
Outstanding Disk I/Os	339
Transaction Log Writes	339
Transaction Log Allocations	339
Session Leaders - I/O	340
I/O Error Rate	340
I/O Detail View	342
Space Statistics - Sybase	345
Database Overview	345
Log Overview	346
Device Overview	347
Space Detail View	348
Databases Statistics - Sybase	351
Database Summary	351
Database Detail View	352
Contention Statistics - Sybase	356
Logical Lock Contention	357
Address Lock Contention	358
Group Commit Sleeps	358
Modify Conflicts	358
Device I/O Contention	358
Disk I/O Structures	359
Server Configuration Limit	359

Engine Configuration Limit	360
Operating System Limit	360
Deadlock Rate	360
Deadlocks	361
Current Locks	361
Lock Contention	362
Blocked	362
Network Contention Rate	363
Network Delays	364
Network Requests	364
Contention Detail View	364
Users Statistics - Sybase	367
Active Connections	368
Inactive Connections	368
Current Locks	368
Users Blocked	369
Committed Transactions	369
ULC Flushes	370
Rows Inserted	370
Rows Updated	370
Rows Updated (Data Only Locks)	371
Rows Deleted	371
Total Rows Affected	371
Session Leaders - Memory	371
Session Leaders - I/O	372
Session Leaders - CPU	372
Users Detail View	373
Network Statistics - Sybase	378
Average Bytes Received Per Packet/Average Bytes Sent Per Packet	379
Total Packets Received	379
Total Packets Sent	379
Total Bytes Received	379
Total Bytes Sent	379
Network Errors	379
Network Contention Rate	380
Network Delays	380
Network Requests	381
OS Page Statistics	381
Summary Tab	381
Processor Time	382

Processor Speed	382
Processor	382
Disk Time	382
Load Average	383
Paged Memory Used	383
Number of Processors	383
Swap Memory Used	383
Average Disk Queue	383
Page Faults/Sec	384
Processor Queue	384
Network Output Queue/Network Queue	384
Available Physical Memory	385
Available Paged Memory	385
Available Swap Memory	385
Total Physical Memory	386
Total Paged Memory/Total Swap Memory	386
Used Disk Space	386
Total Disk Space	386
Free Disk Space	387
Processes Overview	387
Number of Logins	388
Number of Processes	388
Oracle Expert Guide	396
Home Statistics	396
Memory Vital Signs	397
Contention Vital Signs	397
I/O Vital Signs	399
Users Vital Signs	400
Space Vital Signs	402
Network Vital Signs	403
Library Cache Hit Ratio	405
Dictionary Cache Hit Ratio	407
Memory Sort Ratio	408
Parse/Execute Ratio	409
Buffer Busy Waits	409
Rollback Contention Ratio	410
Latch Miss Ratio	410
Parallel Query Busy Ratio	411
Free Shared Pool Percent	411
Problem Tablespaces	412

Problem Objects	413
Top System Bottlenecks	413
Top Session Bottlenecks	414
Current Object Blocks	414
Enqueue Waits	415
Free List Waits	415
Total Used Space/Total Free Space	415
Archive Log	416
Top Processes	416
Active User Processes	416
Inactive User Processes	417
Memory Page Statistics	417
Key Ratio Analysis - Memory	418
Bottleneck Analysis - Memory	418
SQL Analysis - Memory	419
SGA Analysis - Memory	420
Workload Analysis - Top Memory Hogs	421
Buffer Cache Hit Ratio	421
Buffer Pool Usage	422
Buffer Pool Allocations	423
Free Memory and Used Memory in Shared Pool	424
Library Cache Hit Ratio	425
Dictionary Cache Hit Ratio	426
Memory Sort Ratio	427
Free Memory and Used Memory in Shared Pool	428
Session Leaders - Memory	429
SGA	430
Memory Detail	430
I/O Statistics	436
Archive Files Written Today	437
User Object Accesses	438
Active Rollbacks	439
Session Leaders - I/O	439
Redo Wastage and Redo Log Size	440
Rollback Gets and Rollback Waits	440
Physical I/O	441
Logical I/O	441
I/O Detail	442
Space Statistics	448
Tablespace Overview	448

Fragmentation Leaders	449
Schema Leaders - Space	450
Space Detail	450
Objects Page Statistics	457
Object/Buffer Pool Placement	458
Chained Table Placement	459
Active Rollback Transactions	459
Objects With Space Deficits	462
Objects With Extent Problems	462
Objects Pinned	463
Invalid/Unusable Objects	463
Locked Objects	464
Rollback Summary	464
Objects Detail	464
OS Page Statistics	471
Processor Time	472
Processor Speed	472
Processor	473
Disk Time	473
Load Average	473
Paged Memory Used	473
Number of Processors	473
Swap Memory Used	473
Average Disk Queue	474
Page Faults/Sec	474
Processor Queue	474
Network Output Queue/Network Queue	475
Available Physical Memory	475
Available Paged Memory	475
Available Swap Memory	476
Total Physical Memory	476
Total Paged Memory/Total Swap Memory	476
Used Disk Space	476
Total Disk Space	477
Free Disk Space	477
Processes Overview	477
Number of Logins	478
Number of Processes	478
Contention Statistics - Oracle	486
Latch Miss Ratio	487

Latch Immediate Miss Ratio	488
Latch Sleep Ratio	488
Buffer Busy Wait Ratio	489
Rollback Contention Ratio.	489
Users Blocked	490
Users Waiting (General)	491
Sessions Waiting for Latches	491
Sessions in Buffer Busy Waits	491
Prolonged Lock Waits	492
Redo Log Space Requests	492
Redo Log Space Wait Time	493
Parallel Query Busy Ratio.	494
Used Query Slaves	494
Max Query Slaves.	495
Free List Waits	496
Enqueue Waits	496
Contention Detail View	497
Network Statistics - Oracle	502
Bytes Sent to Client.	503
Bytes Received from Client.	503
Roundtrips to/from Client	503
Bytes Sent to DBLink	503
Bytes Received from DBLink	504
Roundtrips to/from DBLink	504
Network Contention.	504
MTS Response Activity	505
MTS Request Activity	505
Users Page Statistics	506
Active Connections	507
Current Locks and Max Open Locks.	507
Current Transactions and Max Transactions	507
Total Connections and Max Connections	508
Inactive Connections.	508
Open Cursors	509
Users Waiting	509
Users Blocked	510
Sessions Involved in Disk Sorts	510
Leading Sessions - CPU.	511
Leading Sessions - Memory	512
Leading Sessions - I/O	512

Users Detail	513
Session Details	519
Session Memory Tab for Oracle	519
Session I/O Tab for Oracle	519
Session Contention Tab for Oracle	520
Session Objects Tab for Oracle	521
Session Network Tab for Oracle	521
Session SQL Tab for Oracle	521
Session Statistics Tab for Oracle	522
DB2 Expert Guide	523
Home View Statistics	523
Memory Vital Signs	523
Contention Vital Signs	523
I/O Vital Signs	524
Users Vital Signs	524
Space Vital Signs	524
Instance Vital Signs	524
Memory Page Statistics	525
Buffer Pool Hit Ratio	525
Buffer Pool Index Hit Ratio	526
Database Heap Utilization	526
Catalog Cache Hit Ratio	526
Sort Overflow Percentage	527
Hash Join Overflow Percentage	527
Catalog Cache Overflows	528
Package Cache Overflows	528
Private Workspace Overflows	528
Shared Workspace Overflows	529
Lock List Utilization	529
Package Cache Hit Ratio	530
Session Leaders – Memory	530
Sort Heap Utilization	530
Shared Workspace Hit Ratio	531
I/O Page Statistics	531
Direct Read Ratio	532
Logical Read Ratio	532
Physical Read Ratio	533
Synchronous Read Ratio	533
Asynchronous Read Ratio	533
Direct Write Ratio	533

Synchronous Write Ratio	533
Asynchronous Write Ratio	533
Log Read Rate	534
Direct Read Rate	534
Synchronous Read Rate	534
Asynchronous Read Rate	534
Log Write Rate	534
Direct Write Rate	534
Synchronous Write Rate	535
Asynchronous Write Rate	535
Prefetchers	535
Page Cleaners	535
Log Space Cleans	535
Dirty Page Cleans	536
Prefetch Wait Time	536
Buffer Pool Hit Ratio	536
Buffer Pool Index Hit Ratio	537
Victim Cleans Ratio	537
Threshold Cleans Ratio	537
Log Space Cleans Ratio	537
Session Leaders – I/O	538
Space Page Statistics	538
DMS Space (Total Used and Total Free)	539
DMS Utilization	539
SMS Space (Total Used and Total Free)	539
SMS Utilization	540
Active Log Size	540
Active Log Used	540
Active Log Free	540
Active Log Utilization	540
Secondary Logs Allocated	540
Secondary Log Used HWM	541
Inaccessible Containers	541
Tablespaces Low on Space	541
Abnormal State Tablespaces	542
Tablespace Overview	542
Object Page Statistics	543
Buffer Pools Tab – Overview	543
Buffer Pools Tab – I/O Analysis	543
Tablespaces Tab – Overview	544

Tablespaces Tab – I/O Analysis	545
Containers Tab	546
Tables Tab	546
Contention Page Statistics	547
Deadlocks	547
Lock Waits.	548
Lock Timeouts.	548
Lock Escalations	548
Applications Waiting on Locks	549
Lock Overview.	549
Lock List Utilization	549
Lock Wait Time	550
Average Lock Wait Time.	550
Users Page Statistics	551
Connections	551
Connections Idle	552
Connections Active	552
Connections Waiting	552
Static SQL Statements	552
Dynamic SQL Statements.	552
Transactions	553
Transactions Per Second	553
Sorts Per Transaction	553
Lock Waits Per Transaction	553
Selects Per Transaction	553
Rows Selected Per Transaction	553
Session Leaders – Memory	553
Session Leaders – I/O.	554
Session Leaders – CPU	554
Max. Connections	555
Percent Executing.	555
Percent of Maximum	555
Users Detail.	555
Instance Page Statistics	560
Sort Heap Utilization	562
Monitor Heap Utilization	562
Sort Heap (MB)	562
Private Memory (MB)	563
Piped Sorts Requested	563
Piped Sorts Rejected	563

Post Threshold Sorts	564
Post Threshold Hash Joins	564
Buffer Utilization	565
Request Block Utilization	565
Message Anchor Utilization	565
Connection Entry Utilization	565
Instance Identification	566
Monitor Switches	566
Idle	567
Stolen	567
Requests	567
Registered	567
Request Overflows	568
Waiting for Tokens	568
Assigned from Pool	568
Created Empty Pool	568
Database Configuration	569
Memory Pools	569
Memory Pool Utilization	569
Memory Pool Size	570
Instance Utilities	570
FCM Throughput	570
FCM Resource Utilization	571
Agent Utilization	571
Percent of Agents Stolen	571
Other Views and Statistics	572
Archive View	572
Health Index View	573
Hot Objects	573
Hot Tables	573
Hot Code	574
Lock View	574
All Locks Tab	575
All User Locks Tab	577
Blocking Locks Tab	578
Locks View for DB2	581
Operating System View	583
OS Page Statistics	584
Summary Tab	585

Processor Time	585
Processor Speed.	586
Processor	586
Disk Time	586
Load Average	586
Paged Memory Used	586
Number of Processors	587
Swap Memory Used	587
Average Disk Queue	587
Page Faults/Sec	587
Processor Queue	588
Network Output Queue/Network Queue	588
Available Physical Memory	588
Available Paged Memory	589
Available Swap Memory	589
Total Physical Memory	589
Total Paged Memory/Total Swap Memory	589
Used Disk Space.	590
Total Disk Space.	590
Free Disk Space	590
Processes Overview	591
Number of Logins	591
Number of Processes	591
Session Detail View	599
Oracle Session Detail View.	599
SQL Server Session Detail View.	603
Sybase Session Detail View	609
DB2 Session Detail View	613
Top Sessions View	622
Memory Tab	622
I/O Tab	626
CPU Tab	630

Top SQL View	632
Glossary	634
A	634
B	634
C	634
D	635
E	635
F	635
G	635
H	636
I	636
J	636
K	636
L	636
M	637
N	637
O	637
P	638
Q	638
R	638
S	638
T	639
U	639
V	639
W	639
X	639
Y	639
Z	639

Welcome to Performance Center

Welcome to Performance Center, a 24/7 performance monitoring solution featuring custom alert thresholds, historical reporting, and comprehensive status information available through a variety of sources, including a web-based user interface that can be accessed through your system's local web browser. Performance Center ensures maximum availability and performance across database platforms by providing automated problem identification facilities that enable the proactive and reactive monitoring, diagnosing, and managing of multiple Oracle, SQL Server, Sybase, and DB2 LUW database environments in real time.

About This Document

The Performance Center User Guide is the primary reference for Embarcadero's Performance Center. It provides an overview of the product and gives detailed instructions on common and less-common tasks. In HTML Help format it serves as the online help for the application. It is also distributed in PDF format for easy downloading and printing.

For licensing and installation information, see the *Performance Center Installation Guide*.

For pointers to additional information not included in this document, see [Additional Product Information](#)

Product Benefits by Audience

DATABASE ADMINISTRATORS

Performance Center is critical for organizations concerned with proactively ensuring that databases are running at peak efficiency, while at the same time providing a way to quickly resolve problems when they do occur. With its intuitive approach to problem determination combined with expert help for every monitored statistic, even intermediate to novice DBAs can become extremely productive. Solving problems using the expert help recommendations, as well as observing the many things that can cause performance degradation, actually serves as a tutorial for less experienced administrators. For the advanced DBA who simply cannot be everywhere at once, Performance Center's customization allows him or her to effectively have a set of virtual eyes on a vast array of databases at one time.

DATABASE ADMINISTRATOR MANAGEMENT

Getting a handle on precisely how well a company's databases are performing is often quite challenging. There is a real tendency for users to think that the databases are always the cause of a performance problem, even when many other factors may be at work. Moreover, defending Service Level Agreements is often impossible due to the lack of real-world data showing things like database availability. With Performance Center's Historical Reports, all this information (and much more) can be easily captured and presented at an executive summary level. This information is valuable not only to management, but also the DBA who may realize that a certain database performed poorly over the past month and may require more attention in the short term.

Additional Product Information

The Embarcadero Web site is an excellent source for additional product information, including white papers, articles, FAQs, discussion groups, and the Embarcadero Knowledge Base.

Go to www.embarcadero.com/support, or click any of the links below, to find:

- [Documentation](#)

- [Online Demos](#)
- [Technical Papers](#)
- [Discussion Forums](#)
- [Knowledge Base](#)

About This Guide

This guide contains the following sections:

Section	Description
Introduction	This section provides an overview of the application and contains information on the following additional topics: Technical Requirements, Licensing, Using Help, and Technical Support.
Application Basics	This section provides conceptual information about Performance Center design, navigation, maintenance dialogs, and views.
Web Client	This section provides information on Performance Center's Web Client interface.
Using Performance Center	This section provides detailed information about using Performance Center.
Oracle Expert Guide	This section provides expert help for all Oracle categories and statistics displayed by Performance Center.
SQL Server Expert Guide	This section provides expert help for all Microsoft SQL Server categories and statistics displayed by Performance Center.
Sybase Expert Guide	This section provides expert help for all Sybase ASE categories and statistics displayed by Performance Center.
DB2 Expert Guide	This section provides expert help for all IBM DB2 categories and statistics displayed by Performance Center.

Opening Performance Center Client

During the Performance Center Server installation, the user who installs the Performance Center Repository is automatically granted the role, PERFCNTR_ADMIN, which grants administrative privileges. To grant administrative privileges to additional users, the DBA should add users to this role.

TIP: You can use Embarcadero DBArtisan to grant roles to additional users.

It is not necessary to log on to a datasource to use Performance Center. However, to perform maintenance on the Performance Center Repository (including registering datasources) you must log on as a user with administrative privileges.

Opening Performance Center

- 1 On the Windows Taskbar, click **Start**, and select **Programs>Performance Center>Performance Center**. Performance Center opens to the Enterprise view and the Administrator Login dialog appears.



- 2 Log on to an Performance Center Server by typing the appropriate user name and password into the fields provided and click **OK**. Alternatively, you can use Performance Center without logging on to a server by clicking **Cancel**.

Additionally:

- Indicate that you want to automatically log on the next time you select the server by choosing the **Auto Login** option.
- De-select **Show this window at startup** to indicate that you do not want the **Administrator Login** to appear when Performance Center is started. This setting can also be configured in the [Options Editor](#).

About Performance Center

Performance Center offers easy access to your license information. In the About Performance Center dialog you can see version and build information, as well as seeing your license type, licensed modules (if any), and server, system, and registration information. You can also connect to numerous Embarcadero Technologies links.

NOTE: From a client machine, you can see information about the licenses available to your particular machine. Only the Embarcadero License Server Administrator dialog, housed on a dedicated server machine, gives an overview of the licenses and servers available across the organization.

The table below describes the information available in the About Performance Center dialog:

Element	Description
Performance Center version and copyright	Displays the version and build number of the current installation along with the copyright years.
License Type	Displays the license type, floating or nodelock. Stand-alone installations are nodelock licenses. Floating licenses require that your local machine be connected to an Embarcadero License Server.

Element	Description
Licensed Modules	Displays a list of the possible application modules and whether they are in use or not.
Server	If you have a floating license, you can see the license server you are connected to. If for some reason you need to connect to a different license server, for example if your enterprise has multiple server machines and the one you are connected to fails, you can ping another server machine and make an alternate connection. NOTE: You can enter and ping the information in the Licensing dialog, but you must restart DBArtisan for the new connection to take place.
System	The System Info dialog displays the IP address for your machine.
License	Opens the License Request Wizard .
Web Links	Makes connections to Embarcadero Technologies available to you, including contact information, downloads, product information, technical support, company information, upgrade requests, and FAQs.

Performance Center Overview

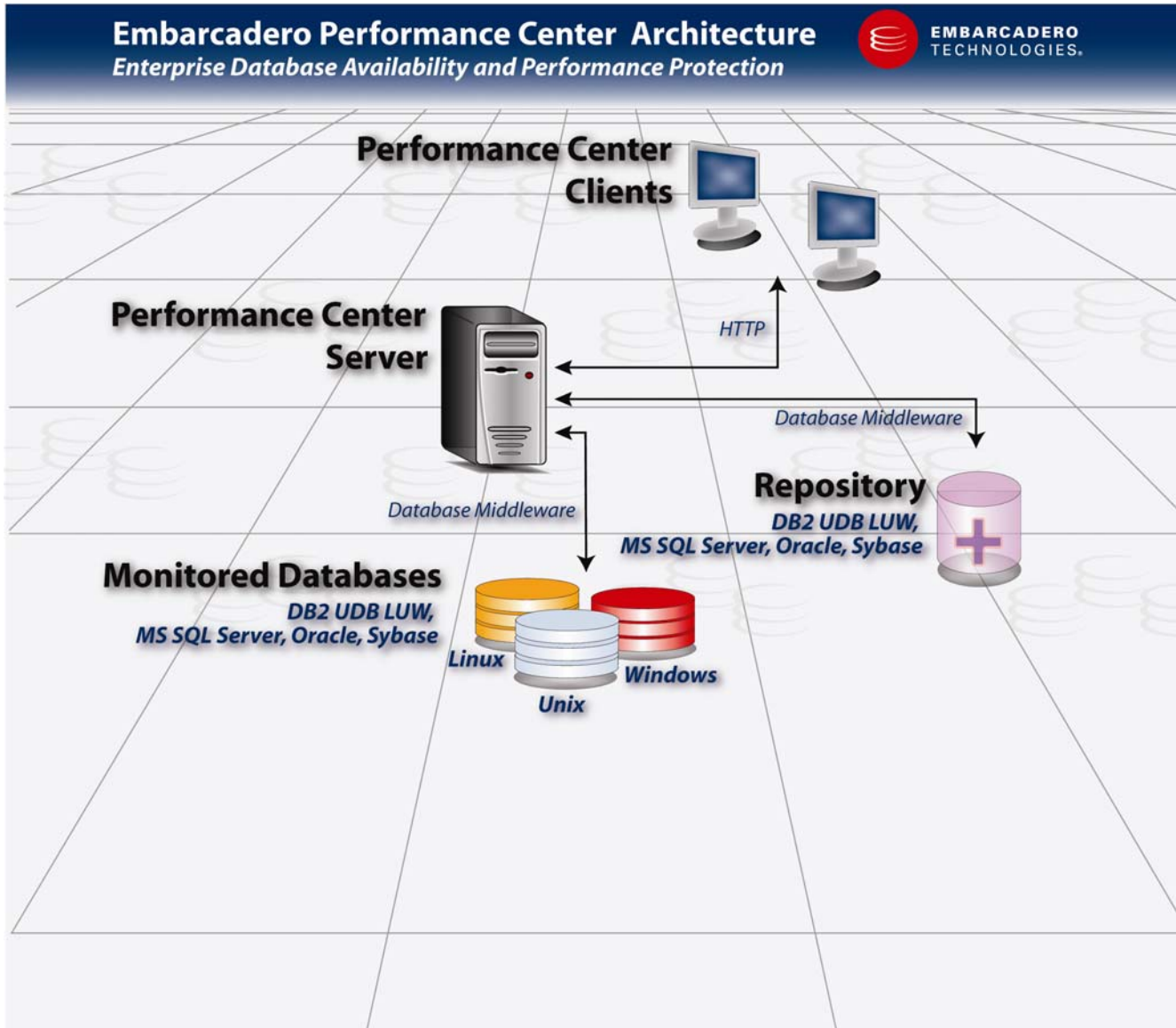
Application Basics is designed to situate you within the application and provide information about what Performance Center offers in the way of design, navigation, maintenance dialogs, and other application features. The information presented here is high-level and conceptual. For information on how to use Performance Center, see [Using Performance Center](#).

Application Basics is divided into six sections, the table below describes each section:

Topic	Description
Architecture	This section demonstrates how the different components of the application communicate.
Product Design	This section describes the User Interface (UI) of the application and includes descriptions about product layout, windows, views, etc., and customizing options.
Product Navigation	This section includes descriptions about the Performance Center menu and toolbar.
Maintenance Dialogs	This section is a high-level overview of the maintenance dialogs.
Configuring Performance Center Client	This section describes customizing the Performance Center Client for your specific needs.

Application Design

Performance Center relies on a server component to handle the actual monitoring of all enterprise databases and also uses a central repository to store the metadata needed for maintaining demographic information and performance analysis. You can install the Performance Center Client, which is the user interface, on one or more machines. You can manage your enterprise. The diagram below demonstrates how the different components of the application communicate.



Application Interface

The Performance Center UI is designed to make monitoring your enterprise an easy task. The UI includes several views for global and specific enterprise monitoring. In addition, the UI includes several other important features to provide you with a wealth of information about your enterprise.

Enterprise Performance Center [Enterprise View]

Datasource Maintenance Report View Window Help

TORLABSQL00_2 (SQL Server 2000) Zoom 06h 00m Current

Datasources

- TORLABSQL00_2 (SQL Server 2000)
- SAMPLE (DB2 09.01.0000)
- TORLABSQL00_1 (SQL Server 2000)

	0 Servers in Healthy State	0.00%	
	0 Servers in Don't Monitor State	0.00%	
	1 Servers in Problem Monitoring State	33.33%	<div style="width: 33.33%; background-color: red;"></div>
	0 Servers in Low State	0.00%	
	0 Servers in Warning State	0.00%	
	2 Servers in Critical State	66.67%	<div style="width: 66.67%; background-color: red;"></div>

Name	Type	Datasource Up	Listener Up	Health Index (%)
TORLABSQL00_2	SQL Server 2000	Yes	N/A	89.10
SAMPLe	DB2 09.01.0000	Yes	Yes	N/A
TORLABSQL00_1	SQL Server 2000	Yes	Yes	90.04

	The datasource is running and connected.		The datasource is running and connected, but there is an error.		Performance Center cannot connect to this datasource.		The datasource is not being monitored.
--	--	--	---	--	---	--	--

Enterprise View x TORLABSQL00_2 - HOME

Message

tion - Mean fetch size: 16120.00 bytes, Mean fetch time: 0.03 secs, Fetch interval: 3.00 secs
 tion - Mean fetch size: 16099.00 bytes, Mean fetch time: 0.01 secs, Fetch interval: 3.00 secs

The Product Design section outlines the following components:

- [Views](#)
- [Workbook](#)
- [Output Window](#)
- [Status Bar](#)
- [Event Log](#)
- [Explorer](#)

Views

Performance Center views are designed to provide you with several different ways to monitor your enterprise. The views include a combination of graphics and a color-coded severity level, where you can quickly determine the health of all registered datasources in your enterprise. Performance Center Views include an [Enterprise view](#) and [Performance Category views](#). The table below describes the Performance Center views:

View	Description
Enterprise	Provides a general overview of the health of all registered datasources.
Performance Category	Includes performance category views, each of which provides more specific details regarding the currently selected datasource. The Home view is the default Performance Category view.

The Enterprise view and Performance Category views use the same color-coded severity level scheme. This enables you to identify problems by entering Performance Category when you see a threshold violation indicated on the Enterprise view. The table below describes the color and associated severity level:

Color	Severity
Green	Low
Yellow	Warning
Orange	High
Red	Critical

Your monitoring requirements determine which view to use. For example, to quickly review all the registered datasources in your enterprise, click the Enterprise tab. To review the performance of a particular datasource, select the target datasource, click the Performance Category view tab, and then navigate to one of the eight performance category views.

NOTE: In addition to the above views, you can customize your view to display the [Explorer](#) and the [Output Window](#).

Enterprise View

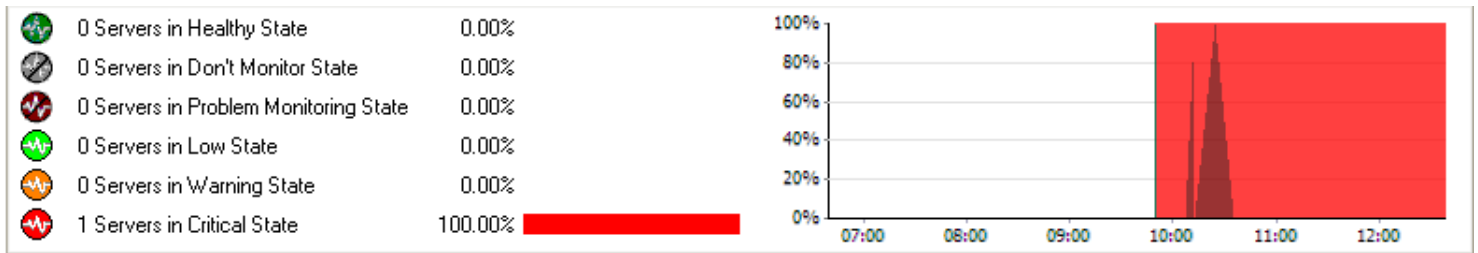
The Enterprise view is a single view designed to give you a comprehensive picture of the overall health of all monitored datasources in your enterprise. It displays all currently registered datasources. In the Enterprise view you can know what databases are up and down and gives you other critical indicators regarding the status of your monitored datasources. The Enterprise view for the main or “fat” client and the Web client differ slightly. The views are described below:

Enterprise View for the Main Client

The table below describes the Enterprise view columns for the main client:

Column	Description
Name	Displays the name of the datasource.
Type	Displays the database platform and version number of the datasource.
Datasource Up	Indicates if the datasource is running.
Listener Up	ORACLE ONLY Indicates if the Oracle-specific datasource listener is functioning.
Health Index	Indicates the current overall performance level of the registered datasource. The Health Index is on a scale of 0-100 with 100 being the ideal.
Alarm	Indicates any alarms for the registered datasource.

In addition to category selections, the **Enterprise** view also contains the **Enterprise View Summary**, a graphical display of the network that Performance Center monitors.



The values on the left display a summary of the current states of the data sources in the network, while the right side of the screen provides the same information in graph format.

Enterprise View for the Web Client




The table below describes the Enterprise view columns on the Web Client:

Column	Description
Datasource	Displays the name of the datasource(s).
DBMS	Displays the datasource platform.
Version	Displays the platform version of each datasource.
Datasource Up	Indicates if the datasource is running.
Listener Up	ORACLE ONLY Indicates if the Oracle-specific datasource listener is functioning.
Health Index	Indicates the current overall performance level of the registered datasource. The Health Index is on a scale of 0-100 with 100 being the ideal numeric representation of overall performance.
Alarms	Indicates any alarms for the registered datasource.

Enterprise View Graphics

In addition to the information in the columns, the Enterprise view combines graphics and a color-coded severity level to indicate the status of all monitored datasources. The table below describes each graphic:

Graphic	Description
	The datasource is running and connected.
	The datasource is running and connected, but there is an error.
	The datasource is running but not connected. NOTE: This icon does not appear in the Web Client.

Graphic	Description
 The datasource is not being monitored.	The datasource is not being monitored.
 The datasource is not running.	The datasource is not running.
 The listener is not running.	The listener is not running.

Performance Center colors each graphic, for both the main Enterprise View and the Web client, to give you a general overview of the health of your datasources and to alert you to any threshold violations. The table below describes the color and associated severity level for your datasources:

Color	Severity
Green	Low
Yellow	Warning
Orange	High
Red	Critical

You can use the Enterprise view to quickly determine the status of all the registered datasources in your enterprise, and then open the [performance category view](#) to pinpoint the exact problem area for the target datasource.

In the Enterprise view, you can right-click any datasource and access the [Datasource menu](#) functionality. You can also right-click anywhere in the Enterprise view to customize the display.

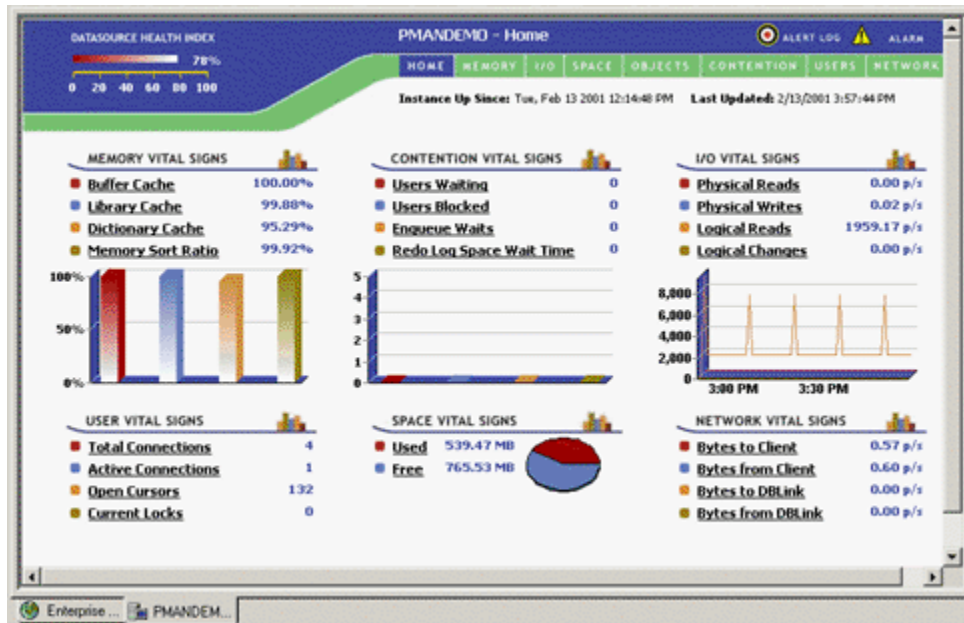
Performance Category Views

The performance category views provide you with important statistics for monitoring your datasources. The overall categories are the same for all supported platforms, however, the information available is specific to the datasource currently selected. You can open more than one performance category view by selecting a datasource either from the [Enterprise](#) view or the [datasource list](#). When you have several category views open, you can navigate to each one by selecting the [Workbook tab](#) for the target datasource.

Every performance category view displays the [Health Index](#) and includes links to the [Alarm Center](#). For Oracle-specific databases, a link to the [Alert Log](#) view is also provided. The default performance category view is the [Home view](#), which gives you an overall picture of the most important performance categories.

Related Topic:

[Performance Center Web Client Views](#)



Each performance category view page displays performance statistics for the target monitored datasource including the server/instance startup time. On every page, you can click any statistic heading, any underlined statistic, the Alarm Center, and for Oracle, the Alert view, to get more detailed information. Depending on what you select, Performance Center displays the performance statistic page or details about the target statistic. All of the graphs and charts on the performance category views are customizable; double-click them to configure them for your needs.

NOTE: This example displays Oracle information. However, you can click any statistic heading, any underlined statistic, the Alarm Center, and for Oracle, the Alert view, to get more detailed information for all supported platforms.

Navigating Performance Category Views

The performance category views include global and specific navigation so you can quickly access different levels of detail depending on your needs. The performance category views include the following navigation options:

- [Performance Category View Menu](#)
- [Performance Category View Shortcut Menus](#)
- [Statistic Shortcut Menu](#)
- [Detail View Shortcut Menu](#)

Customizing Performance Category Views

You can customize any of the charts and graphs on the performance category views by double-clicking a chart or graph to open the Editing dialog. All the graphs and charts on the performance category views are customizable; double-click them to configure them for your needs.

Performance Category View Menu

In the Performance Category View Menu you can navigate to the eight different category views, which give you more specific statistical information about your monitored datasources.

HOME	MEMORY	I/O	SPACE	DATABASES	CONTENTION	USERS	NETWORK
------	--------	-----	-------	-----------	------------	-------	---------

On the menu, you can select any performance category to navigate to that view. The performance category views menu bar is color-coded to indicate the current page as well the state of statistics in any given category. The table below names the color and describes the associated meaning:

Color	Meaning
Blue	Current page.
Green	Statistics in this category are in good health.
Red	Threshold violation in one or more statistics in this category.

The default performance category view is [Home](#).

Performance Category View Shortcut Menus

Each performance category view includes a shortcut menu that you can use to navigate to the detail page view for the target performance category. In addition, the [detail view](#) page includes a shortcut menu that offers more navigation options. To open the shortcut menu, right-click any performance category tab, and then click the target detail statistic.

Statistic Shortcut Menu

The performance categories on every performance category view page include shortcut menus. The table below describes the shortcut menu items:

Menu Item	Description
Details	Opens the detail page for the target statistic.
Details in New Window	Opens a new window containing the detail page for the target statistic.
Alerts	Opens the Datasource Alerts Maintenance dialog.
Properties	Opens the Statistic Properties dialog.
Help	Opens the DBMS-specific Expert Guide to the page that corresponds to the target statistic.

Detail View Shortcut Menu

Each detail view includes a shortcut menu that offers more navigation options. To open the detail view shortcut menu, right-click any performance category tab, and then click the target detail statistic. The table below describes the shortcut menu items.

Menu Item	Description
Copy	Copy any data you highlight in Performance Center and paste it into another application.
Save	Save any data presented in one of Performance Center's grids to an operating system file for later analysis or to be used in another application like Microsoft Excel.
Manage with DBArtisan	Opens Embarcadero DBArtisan, connects to the currently monitored datasource, and navigates to the object selected in Performance Center. NOTE To use this feature, you must have Embarcadero DBArtisan installed on the same client workstation as Performance Center and the datasources used in both Performance Center and Embarcadero DBArtisan must have the same name.

Menu Item	Description
Help	Opens the Expert Guide to the page that corresponds to the target statistic.

Home View

Performance Center opens displaying the Home view, offering you a quick glance at the most important statistics in your environment. The default home view is the first datasource you register. If you have multiple registered datasources, on the [Enterprise view](#), you can select the target datasource to open the corresponding Home view. You can open multiple Home views and navigate through them by clicking the sheet tab.

NOTE: Performance Center offers a unique Home view for each supported database platform.

For more information see:

[Oracle Home View](#)

[SQL Server Home View](#)

[Sybase Home View](#)

Oracle Home View

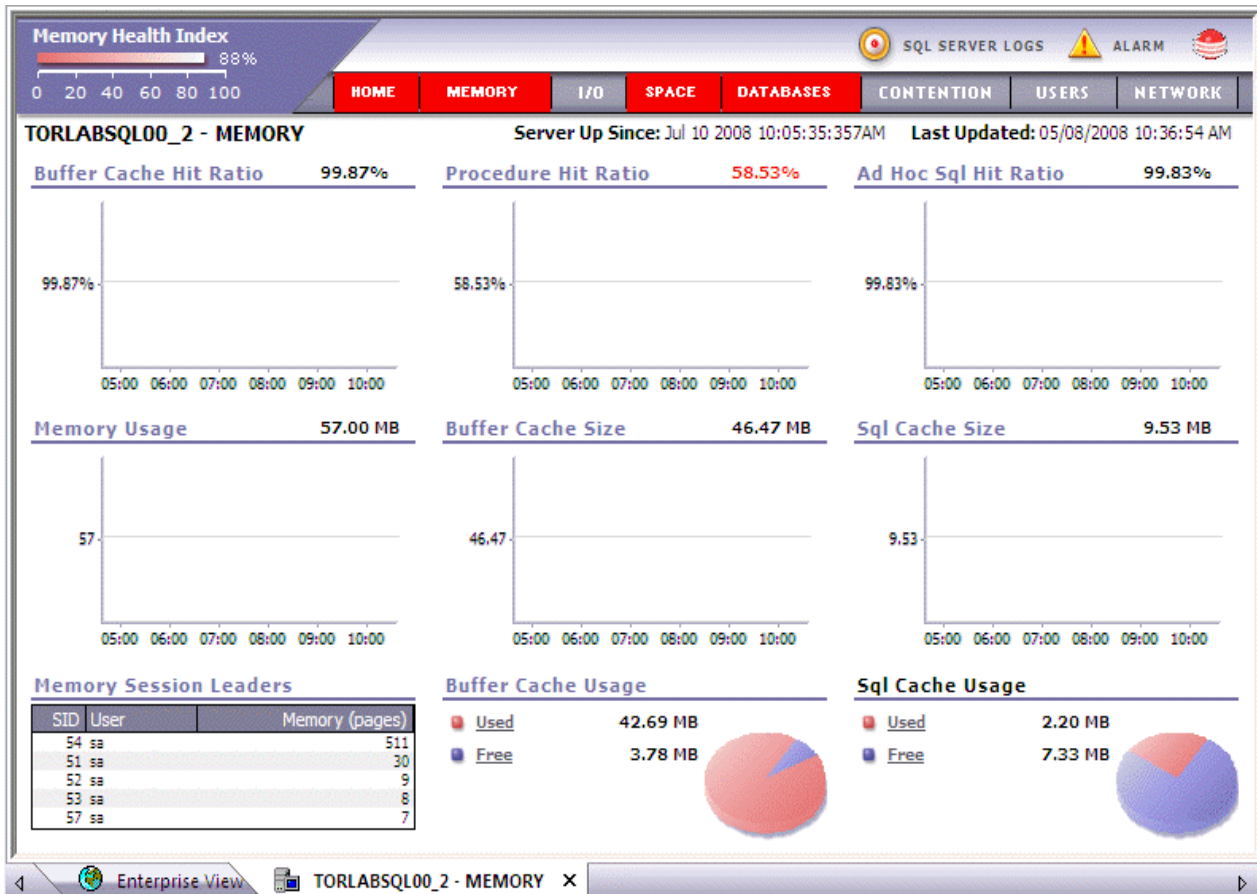
Performance Center opens displaying the Home view, which offers you a quick glance at the most important statistics in your environment. The table below describes each category in the Home view:

Statistic	Description
Memory Vital Signs	Displays summary information about important cache and sort statistics for the current datasource.
Contention Vital Signs	Displays summary information about situations where Oracle and/or users might be in contention.
I/O Vital Signs	Displays summary information about physical and logical datasource Input/Output.
Users Vital Signs	Displays summary information about user statistics.
Space Vital Signs	Displays summary information about used and free space for every storage structure in the current datasource.
Network Vital Signs	Displays summary information about datasource related network activity.

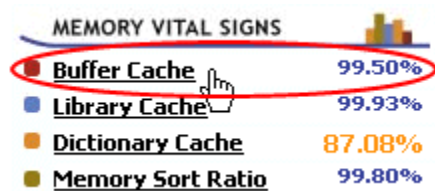
To investigate all of the statistics for a category, click the category title. For example, to view the Buffer Cache information for the Memory Vital Signs category, on the Home view, click Memory Vital Signs:



Performance Center opens the Memory performance category view:



To investigate statistic details, click the statistic to view the drill-down view. For example, to view the Buffer Cache details, on the Home view, click Buffer Cache:



Performance Center opens the Sessions Overview tab of the Memory Detail view.

NOTE: Some statistics do not have a detailed drill-down view. Clicking these statistics opens the target performance category view.

For more information, see [Home View Statistics](#).

Memory - Oracle

The Memory view gives you the most vital statistics for the Oracle SGA, which resides in memory. The Memory page is divided into nine categories. The table below describes these categories:

Statistic	Description
Buffer Cache Hit Ratio	Displays how many times data is found in memory vs. disk.
Buffer Pool Usage	Displays a breakdown of Buffer Pool Usage data.
Buffer Pool Allocations	Displays object placement in caches: Keep, Recycle, Default. This statistic is for Oracle 8 and later.
Library Cache Hit Ratio	Displays how often SQL code is reused by other database users vs. the number of times an SQL statement is broken down, parsed, and then loaded (or reloaded) into the shared pool.
Dictionary Cache Hit Ratio	Displays how often object definitions are found in memory vs. having to read them in from disk.
Free Memory in Shared Pool	Displays how much free memory is in the shared pool and available for reuse.
Memory Sort Ratio	Displays how many sorts are performed in memory and in disk.
Session Leaders - Memory	Displays the top memory session leaders.
SGA	Displays a breakdown of what amount of memory is allocated to system checks.

You can select any of these statistics to open the [Memory Detail](#) view.

For more information on Memory, see [Memory Statistics](#).

I/O - Oracle

The I/O view gives you the most vital statistics for Oracle's Input/Output. It is segmented to display what is going on both at the database and user object level. The I/O view is divided into five categories. The table below describes these categories:

Statistic	Description
Database I/O	Displays amount of Physical and Logical Input/Output activity.
User Object Accesses	Displays overview of user activity against Oracle objects.
Rollback I/O	Displays overview of Rollback Segment Input/Output activity.
Session Leaders I/O	Displays the top I/O session leaders.
Redo I/O	Displays over view of Redo Input/Output activity.

You can select any of these statistics to open the [I/O Detail](#) view.

For more information, see [I/O Statistics](#).

Space - Oracle

The Space view gives you gives you the most vital tablespace statistics for the Oracle server. The Space view is divided into three sections. The table below describes these sections:

Statistic	Description
Tablespace Overview	Displays a space overview of every tablespace.

Statistic	Description
Fragmentation Leaders	Displays the number of free chunks and the largest chunks for each tablespace in the current datasource.
Schema Leaders - Space	Displays top five schema leaders along with their space use.

You can open any of these statistics to open the [Space Detail](#) view.

For more information, see [Space Statistics](#).

Objects - Oracle

The Objects view gives you the most vital statistics for Oracle Server database objects. The Objects view is divided into five sections. The table below describes these sections:

Statistic	Description
Object Summary	Displays summary information about Chained Row Tables, Invalid Objects, Objects With Space Deficits, and Objects Pinned in the current datasource.
User Object Accesses	Displays an overview of user activity against objects.
Rollback Summary	Displays the number of objects at optimal space settings.
Object/Buffer Pool Placement	Displays a count of the number of objects placed in caches: Keep, Recycle, Default. This statistic is for Oracle 8 and later.
Chained Table Activity	Displays the number of times a user has accessed chained row.

You can select any of the these categories to open the [Objects Detail](#) view.

For more information, see [Objects Statistics](#).

Contention - Oracle

The Contention view gives you the most vital statistics regarding situations where Oracle or users might be contending for resources. The Contention view is divided into six categories. The table below describes these categories:

Statistic	Description
System Contention	Displays information about database waits.
Session Contention	Displays information about blocked and waiting user sessions.
Redo Logs	Displays information about how many requests are waiting while the database is flushed and how long these requests are waiting.
Parallel Query	Displays information about how many background processes are running.
Object Contention	Displays Free List and Enqueue Wait information.

You can select any of the these categories to open the [Contention Detail](#) view.

For more information see [Contention Statistics](#).

Users - Oracle

The Users performance category view gives you the most vital statistics for the current datasource users. The Users performance category view is divided into five categories. The table below describes these categories:

Statistic	Description
Session Activity Summary	Displays summary session activity information.
Session Limits	Displays the number of current sessions and their corresponding maximums.
Leading Sessions Memory	Displays the top memory sessions.
Leading Sessions I/O	Displays the top I/O sessions
Leading Sessions CPU	Displays the top CPU sessions.

You can select any of these categories to open the [Users Detail](#) view.

For more information, see [Users Statistics](#).

Network - Oracle

The Network view gives you the most vital network statistics. It displays what is going on between the current datasource and your network. The Network view is divided into five categories. The table below describes these categories:

Statistic	Description
SQL*Net Traffic	Displays the number of SQL trips between the current datasource and your network.
Database Link Traffic	Displays network traffic related to database links.
Network Contention	Displays information about network-related waits.
MTS Response Activity	Displays information regarding the amount of waits in the MTS response queues managed by Oracle's shared dispatcher processes.
MTS Request Activity	Displays information regarding the amount of waits in the MTS request queues managed by Oracle's shared dispatcher processes.

For more information, see [Network Statistics](#).

Other Views and Statistics

Performance Center includes a number of additional views. The table below describes these views:

View	Description
Alarm	The Alarm view displays alarms for all monitored datasources so you can view them. For example, you can set alarms for any threshold violation. When the violation occurs, Performance Center initiates an alarm and logs it for viewing.
Alert Log	You can view a list of your Oracle Alert Logs from the Performance Center application. It includes filters, which let you customize your display for specific Alert Logs
Archive	The Archive view displays lists of all archived log files written out by the database. It pulls information from the database data dictionary and gives you a listing of what happened and when.
Health Index	The Health Index view is a historical display of all Health indexes for the current datasource. It is a time-based view of all the Health Index. You can see all Health Index information, over a certain period of time, simultaneously.
Hot Objects	The Hot Objects view displays all the hot code and hot tables for monitored Oracle datasources.

View	Description
Instance Parameter	The Instance Parameters view displays a list of all Oracle datasource configuration parameters for the current datasource.
Locks	The Locks view displays all active lock information for the current datasource.
Top Sessions	The Top Sessions view displays all sessions in the current datasource. It displays all sessions ranked in I/O or memory priority and orders them by top session hogs.
Top SQL	The Top SQL view displays all SQL running or executed in the current datasource.
Trends	The Trends view is available to let you track statistics for target datasources simultaneously. This is especially useful for determining enterprise trends.
Operating System	The Operating System view displays Windows operating system statistics.

For more information, see the following entries in the Expert Guide:

[Archive View](#)

[Health Index View](#)

[Hot Objects View](#)

[Locks View](#)

[Top Sessions View](#)

[Top SQL View](#)

[Operating System View](#)

SQL Server Home View

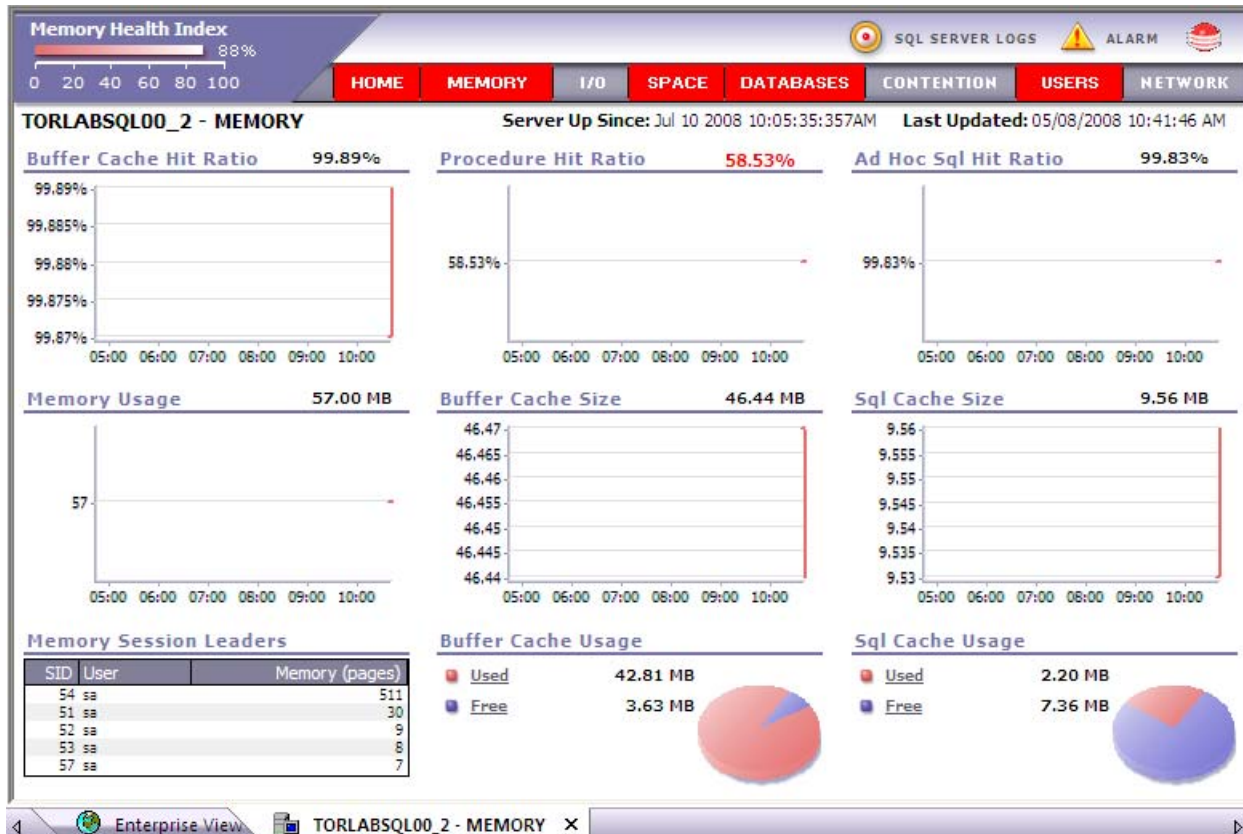
Performance Center opens displaying the Home view, which offers you a quick glance at the most important statistics in your environment. The table below describes each category in the Home view:

Statistic	Description
Memory Vital Signs	Displays summary information about important data and SQL cache, cache hit ratios, and overall memory usage for the current datasource.
Contention Vital Signs	Displays summary information about contention activities occurring at the user and system levels.
I/O Vital Signs	Displays summary information about physical I/O information for databases, database logs, and the server.
Users Vital Signs	Displays summary information about user connections and activity.
Space Vital Signs	Displays summary information about what databases and/or logs are running low on free space.
Network Vital Signs	Displays summary information about datasource related network activity.

To investigate all of the statistics for a category, click the category title. For example, to view the Buffer Cache information for the Memory Vital Signs category, on the Home view, click Memory Vital Signs:



Performance Center opens the Memory performance category view:



To investigate statistic details, click the statistic to see the drill-down view. For example, to view the Buffer Cache details, on the Home view, click Buffer Cache Hit Ratio:



Performance Center opens the Buffer Cache tab of the Memory Detail view:

NOTE: Some statistics do not have a detailed drill-down view. Clicking these statistics opens the target performance category view.

For more information, see [Home View Statistics](#).

Memory - SQL Server

The table below describes these categories:

Statistic	Description
Buffer Cache Hit Ratio	Displays the percentage of requests satisfied from memory as opposed to going to disk.
Procedure Hit Ratio	Displays the percentage of query plan requests generated by stored procedures that are found in the procedure cache area as opposed to going to disk.
Ad Hoc SQL Hit Ratio	Displays the percentage of times that a query plan for an ad hoc SQL statement is found in the procedure cache area as opposed to going to disk.
Memory Usage	Displays the amount of SQL Server memory currently allocated to SQL Server.
SQL Cache Usage	Displays the percentage of the SQL Cache area currently being used.
SQL Cache Hit Ratio	Displays the percentage of log requests satisfied from memory as opposed to going to disk.
Memory Session Leaders	Displays the top sessions with respect to memory consumption.

You can select any of these statistics to open the [Memory Detail](#) view.

For more information on Memory, see [Memory Statistics](#).

I/O - SQL Server

The table below describes these categories:

Statistic	Description
Page Reads	Displays the number of physical database page reads that are issued per second by SQL Server.
Page Writes	Displays the number of physical database page writes that are issued per second by SQL Server.
Read Ahead Pages	Displays the number of physical database page reads in anticipation of use by SQL Server.
Checkpoint Pages	Displays the number of pages flushed to disk per second.
Log Flushes	Displays the total number of log pages for all databases written to disk by the log writer process.
Log Cache Reads	Displays the number of reads performed per second through the log manager cache.
Full Scans	Displays the total number of full table or index scans per second.
Range Scans	Displays the total number of qualified range scans through indexes per second.
Index Searches	Displays the total number of index searches per second.
Probe Scans	Displays the total number of scans, per second, used to directly find rows in an index or base table.
Worktables Created	Displays the total number of work tables created per second.
Forwarded Record Fetches	Displays the total number of records per second fetched by reading forwarded record pointers.
Server I/O Busy Rate	Displays the time that the database server has spent performing input and output operations since the last refresh in Performance Center, in milliseconds.
Session Leaders - I/O	Displays the top sessions with respect to I/O activity.

Statistic	Description
Total I/O	Displays the total number of I/O errors during read and write operations encountered by the SQL Server since the last Performance Center refresh interval.
Total Errors	Total Errors- the total number of I/O errors (errors during read and write operations) encountered by the SQL Server since the last Performance Center refresh interval.

You can select any of these statistics to open the [I/O Detail](#) view.

For more information, see [I/O Statistics](#).

Space - SQL Server

The table below describes these sections:

Statistic	Description
Database Overview	Displays details about the space situation for each database in a SQL Server, including each one's total, used, and free space.
Log Overview	Displays details concerning each transaction log in SQL Server and includes their total, used, and free space amounts.
Disk Free Space	Displays the amount of physical free space available on each of the SQL Server's physical hard drives.

You can open any of these statistics to open the [Space Detail](#) view.

For more information, see [Space Statistics](#).

Databases - SQL Server

The table below describes these sections:

Statistic	Description
Databases	The number of databases defined to the SQL Server instance.
Files	The number of database and log files defined to the SQL Server instance.
File Groups	The number of file groups (containing both database and log files) defined to the SQL Server instance.
Databases w/o Auto-Growth	The total number of databases defined on SQL Server that do not have at least one of their underlying files configured to automatically grow if more space is needed.
Logs w/o Auto-Growth	The total number of database logs defined on SQL Server that do not have at least one of their underlying files configured to automatically grow if more space is needed.
DBs w/o Auto-Creat Stats	The total number of databases defined on SQL Server that do not have the AUTO_CREATE_STATISTICS option enabled.
Errors in Current Error Log	The number of errors that exist in the current SQL Server error log.
Suspect Databases	The number of databases SQL Server has marked as suspect.
Offline Databases	The number of databases SQL Server has marked as offline.
Databases Needing Backup	The number of databases in SQL Server that have not been backed up for more than seven days (excluding the pubs, tempdb, Northwind, msdb, and model databases).
Database Summary	The database overview information on each of the databases defined to SQL Server.

You can select any of the these categories to open the [Database Detail](#) view.

For more information, see [Databases Statistics](#).

Contention - SQL Server

The table below describes these categories:

Statistic	Description
Blocked Users	The number of users currently being blocked in the SQL Server instance.
Page Deadlocks	The number of deadlocks per second detected by SQL Server.
Latch Waits	The number of latches per second that could not be immediately satisfied by SQL Server.
Log Flush Waits	The number of commits per second waiting for a log flush.
Table Lock Escalations	The number of times locks on a table were escalated.
Lock Overview	Displays the total number of acquired locks and blocking locks being experienced per database.
CPU Busy	The time that the CPU has spent working since the last time Performance Center refresh interval, in milliseconds.
I/O Busy	The time that SQL Server has spent performing input and output operations since the last Performance Center refresh interval, in milliseconds.
Server Idle	The time that SQL Server has spent idle since the last Performance Center refresh interval, in milliseconds.

You can select any of the these categories to open the [Contention Detail](#) view.

For more information see [Contention Statistics](#).

Users - SQL Server

The table below describes these categories:

Statistic	Description
Total Connections	The number of user and system processes currently reported in the SQL Server instance.
Active Connections	The total number of active and open threads reported in SQL Server.
Inactive Connections	The total number of inactive and open threads reported in SQL Server.
System Connections	The total number of open threads reported in SQL Server that are owned by system processes.
Users Blocked	The total number of users who are currently being blocked by other processes.
Active Transactions	The total number of active transactions in the SQL Server instance that have not yet been committed, or are waiting on a blocking lock to complete.
SQL Compilations	The number of SQL compilations performed per second, indicating the number of times the compile code path is entered. This also includes compiles due to recompiles.
SQL Re-Compilations	The total number of recompiles triggered per second in a SQL Server instance.
T-SQL Batches	The number of batch requests per second that are processed by the SQL Server instance.
Auto-Param Attempts	The number of auto-parameterization attempts per second and includes failed, safe, and unsafe auto-parameterizations.
Failed Auto-Param Attempts	The number of failed auto-parameterization attempts per second.

Statistic	Description
Session Leaders - Memory	The top users in the SQL Server instance with respect to memory consumption.
Session Leaders - I/O	The top users in the SQL Server instance with respect to I/O activity.
Session Leaders - CPU	The top users in the SQL Server instance with respect to CPU utilization.

You can select any of these categories to open the [Users Detail](#) view.

For more information, see [Users Statistics](#).

Network - SQL Server

The table below describes these categories:

Statistic	Description
Packets Received	The count of all incoming packets received over the network since the last Performance Center sampling.
Packets Sent	The count of all outgoing packets received over the network since the last Performance Center sampling.
Network Reads	The number of network reads performed by SQL Server since the last Performance Center sampling.
Network Writes	The number of network reads performed by SQL Server since the last Performance Center sampling.
Network Bytes Read	The number of bytes that SQL Server read from the network since the last Performance Center sampling.
Network Bytes Written	The number of bytes that SQL Server wrote to the network since the last Performance Center sampling.
Packet Errors	The count of all packet errors that occurred since the last Performance Center sampling.
Total Traffic	The total number of packets that SQL Server received and sent over the network since the last Performance Center sampling.
Network I/O Wait Requests	The total number of times that SQL Server generated an I/O wait request since the last Performance Center sampling.
Network I/O Wait Time (ms)	The number of milliseconds that SQL Server remained in a wait state due to Network I/O Wait Requests since the last Performance Center sampling.

For more information, see [Network Statistics](#).

Other Views and Statistics

Performance Center includes a number of additional views. The table below describes these views:

View	Description
Alarm	In the Alarm view you can get alarms, for all monitored datasources, so that you can view them. For example, you can set alarms for any threshold violation. When the violation occurs, Performance Center initiates an alarm and logs it for viewing.
Health Index	The Health Index view is a historical display of all Health indexes for the current datasource. It is a time-based view of each configured Health Index. In this view you can view all Health Index information, over a certain period of time, simultaneously.

View	Description
Server Configuration	The Server Configuration view displays a list of all SQL Server/Sybase datasource configuration parameters for the current datasource.
Locks	The Locks view displays all active lock information for the current datasource.
Top Sessions	The Top Sessions view displays all sessions in the current datasource. It displays all sessions ranked in I/O or memory priority and orders them by top session hogs.
Top SQL	The Top SQL view displays all SQL running or executed in the current datasource.
Trends	The Trends view is available to let you track statistics for target datasources simultaneously. This is especially useful for determining enterprise trends.
Operating System	The Operating System view displays Windows operating system statistics.
SQL Server Logs	You can view a list of your SQL Server Logs from the Performance Center application. It includes filters, which let you customize your display for specific logs.

For more information, see the following entries in the Expert Guide:

[Health Index View](#)

[Locks View](#)

[Operating System View](#)

[Top SQL View](#)

[Top Sessions View](#)

Sybase Home View

Performance Center opens displaying the Home view, which offers you a quick glance at the most important statistics in your environment. The table below lists each category in the Home view:

Statistic	Description
Memory Vital Signs	Displays summary information about important data and procedure cache statistics for the current datasource.
Contention Vital Signs	Displays summary information about contention activities occurring at the user and system levels.
I/O Vital Signs	Displays summary information about physical I/O information for databases, database logs, and the server.
Users Vital Signs	Displays summary information about user connections and activity.
Space Vital Signs	Displays summary information about what databases and/or logs are running low on free space.
Network Vital Signs	Displays summary information about datasource related network activity.

To investigate all of the statistics for a category, click the category title. For example, to view the Buffer Cache information for the Memory Vital Signs category, on the Home view, click Memory Vital Signs:



Performance Center opens the Memory performance category view.

To investigate statistic details, click the statistic to view the drill-down view. For example, to view the Buffer Cache details, on the Home view, click Buffer Cache:



and then Performance Center opens the Cache Activity tab of the Memory Detail view:

NOTE: Some statistics do not have a detailed drill-down view. Clicking these statistics opens the target performance category view.

For more information, see [Home View Statistics](#).

Memory - Sybase

The Memory view gives you the most vital statistics for the memory areas devoted to holding Sybase data and procedures. The Memory page is divided into five categories. The table below describes these categories:

Statistic	Description
Data Cache Activity	Relays the efficiency at which Sybase is using its allocated data caches.
Procedure Cache Activity	Displays pertinent statistics regarding the activity of the procedure cache.
Large I/O	If large I/O is configured, this area relays if large I/O is being used efficiently
Session Leaders - Memory	Displays sessions using the most memory.
Cache Allocations	Displays the total of all Sybase memory areas as well as breakdowns of the data and procedure caches.

You can select any of these statistics to open the [Memory Detail](#) view.

For more information, see [Memory Statistics](#).

I/O - Sybase

The I/O view gives you the most vital statistics for Sybase's I/O activity. It is segmented to display what is going on both at the database, log, and server level. The I/O view is divided into five categories. The table below describes these categories:

Statistic	Description
Device Activity	Indicates the efficiency of device activity including wait activity, which may degrade overall response times.
Log Activity	Indicates how busy the transaction logs for the server are.
Server Busy I/O Rate	Relays how busy the server is with respect to performing I/O functions.
Session Leaders I/O	Displays users with most current physical I/O usage
I/O Error Rate	Indicates the rate at which the server is experiencing I/O related errors.

You can select any of these statistics to open the [I/O Detail view](#).

For more information, see [I/O Statistics](#).

Space - Sybase

The Space view gives you the most vital space-related statistics for the Sybase server. The Space view is divided into three sections. The table below describes these sections:

Statistic	Description
Database Overview	Displays the amounts of total, used, free and percent free for each database configured on the server.
Log Overview	Displays each transaction log for a database along with its total, used, free, and percent free. NOTE: Databases that have their transaction log on the same device as the database will show the same total and free space amounts for both the database and transaction log.
Device Overview	Displays total, used, free, and percent free for database devices.

You can open any of these statistics to open the [Space Detail view](#).

For more information, see [Space Statistics](#).

Databases - Sybase

The Databases view displays a summary of the activity and current status of all databases configured on the server. Information includes the total I/O, CPU, and memory usage of each database, plus data on whether the database is corrupt or contains corrupt objects.

You can select Database Summary to open the [Database Detail view](#).

For more information, see [Databases Statistics](#).

Contention - Sybase

The Contention View displays summary information about contention activities occurring at the user and system levels. The Contention view is divided into five categories. The table below describes these categories:

Statistic	Description
System Contention	Indicates if contention is occurring at the system level.
I/O Contention	Indicates where I/O contention is occurring.
Deadlocks	Displays a count of deadlocks in relation to total lock requests.
Lock Contention	Displays a count of blocked lock requests in relation to total lock requests.
Network Contention	Displays a count of network delays to total network requests.

You can select any of the these categories to open the [Contention Detail view](#).

For more information see [Contention Statistics](#).

Users - Sybase

The Users performance category view gives you the most vital statistics regarding user activity on the Sybase server. The Users performance category view is divided into five categories. The table below describes these categories:

Statistic	Description
Session Activity Summary	Displays summary session activity information.
Session DML Activity	Displays summary session DML activity.
Leading Sessions Memory	Displays the top memory sessions.
Leading Sessions I/O	Displays the top I/O sessions
Leading Sessions CPU	Displays the top CPU sessions.

You can select any of the these categories to open the [Users Detail view](#).

For more information, see [Users Statistics](#).

Network - Sybase

The Network view gives you the most vital statistics regarding network activity in the Sybase server. The Network view is divided into four categories. The table below describes these categories:

Statistic	Description
Network Packet Summary	Displays the total TDS packets received and sent.
Network Data Summary	Displays the number of bytes received and sent by the Sybase Server.
Network Contention	Displays a count of network errors to the total network requests.
Network Requests	Displays a count of network delays to the total network requests.

For more information, see [Network Statistics](#).

Other Views and Statistics

Performance Center includes a number of additional views. The table below describes these views:

View	Description
Health Index	The Health Index view is a historical display of all Health indexes for the current datasource. It is a time-based view of each configured Health Index. This view displays view all Health Index information, over a certain period of time, simultaneously.
Server Configuration	The Server Configuration view displays a list of all SQL Server/Sybase datasource configuration parameters for the current datasource.
Locks	The Locks view displays all active lock information for the current datasource.
Top Sessions	The Top Sessions view displays all sessions in the current datasource. It displays all sessions ranked in I/O or memory priority and orders them by top session hogs
Operating System	The Operating System view displays Windows operating system statistics.

For more information, see the following entries in the Expert Guide:

[Health Index View](#)

[Locks View](#)

[Top Sessions View](#)

[Operating System View](#)

Performance Center Web Client Views

The performance category views for the Web Client provide you with important statistics for monitoring your datasources. The overall categories are the same for all supported platforms, however, the information available is specific to the datasource currently selected. You can open more than one performance category view by selecting a datasource either from the [Enterprise](#) view.

In addition to the Enterprise View, the Web Client offers a [Monitor Details](#) view, a [Server Details](#) view, a [Client Details](#) view, and an [Admin Login](#) view.

Monitor Details

The Monitor Details view of the Web client where you can see performance details of all registered datasources at a glance. The table below describes the information available to you from this view:

Column	Description
Monitored Datasource	You can see the name of the datasources registered to the server you are monitoring.
Monitor	You can see the monitoring capability and whether or not the datasource is fully enabled or disabled. You enable or disable datasource monitoring from the Datasource Properties dialog.
Status	Shows the current datasource status and whether or not it is OK. Other status possibilities are Error, Unknown, or Transition, but these are rarely, if ever seen.
Connection Detail	See that status of the connection of the datasource to your server. Ideally it is connected and deployed. Other possibilities are Connected/Deployed Failed, Not Connected, Connected/Deployed Execution Error, and so on.
User ID	Displays the user ID.
Query Timeout (seconds)	Displays the amount of time, in seconds, before a query times out. The default is 300.

Column	Description
Ping Interval (seconds)	Displays the amount of time between pings from the server to the datasource. The default is 300.
Primary Interval (seconds)	Displays see the amount of time between primary data sampling intervals. The default is 300.
Secondary Interval (seconds)	Displays the amount of time between secondary data fetch intervals. The default is 900 seconds.
Last Ping Time	The date and time the datasource was most recently pinged are displayed.
Last Primary Fetch Time	You can see the date and time, and the amount of time it took to fetch the primary data most recently.
Last Secondary Fetch Time (seconds)	You can see the date and time, and the amount of time it took to fetch the secondary data most recently.
DBMS	The database platform for the target datasource is displayed.
Version	The database platform version for the target datasource is displayed.

Server Details

In the Server Details view of the Web Client you can see information about the server you are currently connected to.

TIP: To connect to another server, you can use the [Select Performance Center Server](#) dialog or you can type the server name in the Web Client URL:
http://pc_host/perfcntr?ServerManager.dll?RequestPage?Page=default
 where “pc_host” equals the name or IP of the machine hosting the Performance Center server. You can open multiple windows to see details for more than one server simultaneously.

NOTE: To start or stop the target server from this page, go to the Admin Login view and enable your administrator status.

The table below describes the information available from this view:

Column	Description
Server Name	Displays the name of the server you are currently connected to.
Port	Displays the server's port number. The default is 80.
Web Server	Displays the web server. Currently only Apache 2.2.6 is available.
Status	You can see the server's status. The options are OK, Error, Unknown, or Transition.
Status Detail	States whether the server is started or not.
Jobs	Displays the number of jobs currently running on the server.
Repository Datasource	Displays the name of the datasource where the repository is stored.
Status	Displays the repository status and whether it is OK or not.
Connection Detail	Displays the connection status of the repository.
User ID	The name of the repository is displayed.
DBMS	The database platform for the repository is displayed.
Version	Displays the database platform version of the repository.

Client Details

The Client Details view of the Web Client gives you information about the client machine you are currently connected to. The table below describes the information you see:

Column	Description
Client ID	This is the number of fat clients monitoring the same server.
Machine Name	Displays the name of the client machine. In those cases where the server and the client are on the same machine, you will see localhost.
IP Address	Displays the IP address of the client machine.
Jobs	The number of jobs running on the client machine are listed.
Msgs In	The number of messages sent from the server to the client.
Msgs Out	The number of messages sent by the client to the server.
Msg Queue Length	Displays how many messages are waiting in the queue.
Msgs Lost	Shows how many messages were lost. Client can lose messages because the Performance Center Server is producing too much data. If this happens, you can decrease the Polling Frequency to 2 or 1 second(s). However, decreasing the Polling Frequency causes the Performance Center Client to consume more CPU.

Admin Login

Use the Admin Login to sign in as an administrator, if that status is available to you. Simply type your user name and password and click Authorize to enable Administrator Status.

When you are logged in as an administrator, you can start or stop the server from the Server Details page.

Workbook

The Workbook is the Performance Center default display. Performance Center organizes the workspace into a workbook containing a sheet for the [Enterprise](#) view, [performance category views](#), the [Server Manager](#), and any [reports](#). The name of what you selected appears on the tab, letting you distinguish between sheets when you display several of them. For example, if you select the Enterprise view, the label for the workbook tab is Enterprise. If have the Enterprise view and one or more performance category views open, Performance Center displays a workbook sheet labeled Enterprise and one or more workbooks sheets labeled with the datasource name and the specific performance category. To move from sheet to sheet, click the sheet tabs.

NOTE: You can disable the workbook on the [View](#) menu.

Output Window

The Output Window provides vital information about the Performance Center Client, Performance Center Server, and its monitored datasources. The Output Window includes the following tabs:

- [Status](#)
- [Alerts Events](#)
- [Notifications](#)

Status Tab

The Status tab of the Output Window displays messages and a time stamp for information from the Performance Center Server and the Performance Center Client. You can size and sort the columns on the Status tab. You can [clear](#), [copy](#), and [save](#) messages on the Status tab of the Output Window.

Enterprise Alerts Tab

The Enterprise Alert tab of the Output Window displays alert information for all monitored datasources. You can size and sort the columns on the Enterprise Alerts tab. You can [clear](#), [copy](#), and [save](#) information. The table below describes the columns on this tab:

Column	Description
Datasource	The datasource with a statistic that has exceeded a threshold.
Category	The type of statistic.
Description	The statistic that has exceeded a threshold.
Current State	The condition, cleared or not cleared, of the threshold.
Current Value	The numerical value of the statistic.
Severity	The color-coded level of the threshold violation.
Time Alert Started	The time at which the statistic violated the threshold.
Last Time Value Changed	

Notifications Tab

The Notifications tab of the Output Window displays notification information for all monitored datasources. You can size and sort the columns on the Notifications tab. You can [clear](#), [copy](#), and [save](#) information on the Notifications tab of the Output Window. The table below describes the columns on the Notifications tab:

Column	Description
Datasource	The datasource with a notification.
Category	The type of statistic.
Statistic	The statistic that has exceeded a threshold.
Notification Name	The name of the notification.
Time	The time at which the notification was sent.

Working with the Output Window

You can use the Output Window to manipulate information. You can also configure the output window based on your current needs. You can:

- [Clear](#), [copy](#), [save](#) messages.
- [Show](#), [hide](#), [move](#) the window.

Clearing Output Window Messages

- In the **Output Window**, right-click, and then click **Clear**.

Performance Center clears all of the Server Messages.

Copying Output Window Messages

- 1 Right-click the target server messages, and then click **Copy**.

Performance Center copies the target text to the Microsoft Windows Clipboard.

- 2 Paste the contents of the clipboard into the target application.

Saving Output Window Messages

- 1 In the **Output Window**, right-click, and then click **Save**.

Performance Center opens the Save As dialog.

- 2 In the **File Name** box, enter the location and name of the file.

NOTE: Performance Center defaults the file name extension to msg.

- 3 Click **OK**.

Showing the Output Window

- On the **View** menu, click **Output**.

Performance Center displays the Output Window.

Hiding the Output Window

Performance Center offers two ways to hide the Output Window:

- 1 On the **View** menu, click **Output**.

OR

- 2 In the upper left corner of the **Output Window**, click the **x**.

Moving the Output Window

- Drag the move handles or the title bar to the target location.

NOTE: To dock the Output Window, drag it to the edge of the application window.

Status Bar

The Status Bar is the bar at bottom of the application window that displays a brief explanation of the currently selected menu command. The explanation is a short text message about the current condition of Performance Center. The Status Bar also indicates the name and status of the Performance Center Server.

Event Log

The Event Log displays information about system, hardware, and software problems, and security events. When you select the Event Log on the View menu or Performance Center toolbar, the application opens the Performance Center Message Viewer. In the Performance Center Message Viewer you can gather information about the Performance Center Server and Client. You can save the Event Logs as pbw files.


For detailed information on the Performance Center Message Viewer, see the Performance Center Message Viewer Help. After opening the Performance Center Message Viewer, press F1 to access the Performance Center Message Viewer Help.

Opening the Performance Center Message Viewer to View the Event Log

Performance Center offers three ways to open the Performance Center Message Viewer:

- 1 On the **View** menu, click **Message Viewer**.

OR

- 2 On the toolbar, click .

OR

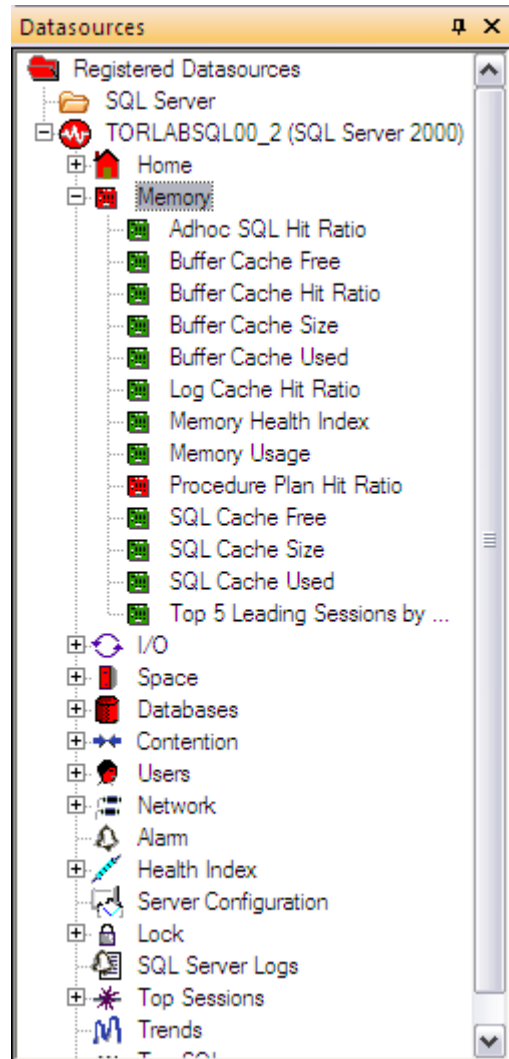
- 3 On the Windows task bar, click **Start > Programs > Embarcadero Performance Center > Message Viewer**.

TIP: When contacting Embarcadero Technical Support, please send the Performance Center Event Log (*.pbw file).

Explorer

The Explorer is a separate, dockable window containing two tabs. The table below describes the two tabs:

Tab	Description
Explorer	The Explorer displays a tree object that you can select and expand. The tree object organizes and nests subjects as branches. By expanding or collapsing the tree, you can efficiently browse multiple datasources and drill down to the appropriate level of detail. You configure the display of datasources on the Explorer. For information, see Configuring the Explorer . For information on datasource groups and profiles see, Organizing Datasources .



Configuring the Explorer

You can configure the Performance Center Explorer tab display to:

- [Show](#)
- [Hide](#)
- [Move](#)

Showing the Explorer

- Click **View > Explorer**. You can choose to show Datasources or Favorites or both. When you select both, the Explorer has tabbed views of each.

Hiding the Explorer

You have two ways to hide the Explorer:

- 1 Click **View > Explorer**. If Datasources, Favorites, or both, are selected, click to deselect.
- OR
- 2 In the top left corner of the **Explorer**, click the **x**.

Moving the Explorer

- Click the title bar and while holding down the mouse, move the Explorer.

TIP: To dock the Explorer, double-click the title bar.

Product Navigation

Performance Center offers three levels of navigation throughout the UI. The table below describes the various ways to navigate through the application:

Navigation	Description
Menu Bar	The menu bar provides the most comprehensive access into the Performance Center application. It includes seven menus that offer complete access to all Performance Center commands.
Shortcut Menus	The shortcut menus let you quickly access the most frequently used commands for each component of the application. They also open the Expert Guides, which offer detailed information about statistics.
Toolbar	The toolbar offers basic functionality, where you can navigate through the performance category views, open the event log, select an Performance Center Server, and select a monitored datasource.

For information on navigating through the performance category views, see [Views](#).

Menu Bar

The Performance Center menu bar offers the most comprehensive access into the application. It contains seven drop-down menus, which offer complete access to all Performance Center commands. You can click the name of any menu to open the list and display the available list of commands and submenus. Many of the menu commands have icons next to them letting you quickly associate the command with the image. Several of the menu icons are also on the [toolbar](#). The Performance Center menu bar includes the following menus:

- [Monitor](#)
- [Datasource](#)
- [Maintenance](#)
- [Report](#)
- [View](#)
- [Window](#)
- [Help](#)

Monitor Menu

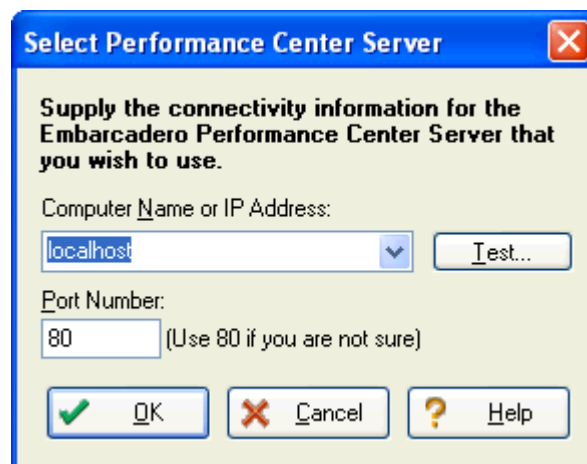
The Monitor menu provides commands that let you interact with your server along with commands that let you set overall configuration options for your Performance Center Server and Client. In the Monitor menu you can:

- [Select a server](#)
- [Start, pause, stop, and ping the current server](#)
- [Configure Performance Center Client using the Options Editor](#)
- [Configure Performance Center Server using the Server Options Editor](#)
- [Customizing the Look and Feel of Performance Center](#)
- [Login/Logout from an Performance Center Server](#)
- [Exit Performance Center](#)

Selecting a Server

In the Selecting a Server command you can select an Performance Center Server.

- 1 To select a new server, click **Monitor > Select**. The Select Performance Center Server dialog appears.



- 2 In the **Computer Name or IP Address** list, type the name or IP address of the computer where the Performance Center Server is running or click the list and then click the name or address.
- 3 In the **Port Number** box, type the port where the Performance Center Server is listening to requests from clients.

NOTE: Performance Center defaults to port 80 (the standard port for the HTTP protocol).
- 4 To test the connectivity to the specified Performance Center Server, click **Test**. An ICMP echo request is sent to the server and then opens a Test Result message box with the response from the server.
- 5 Click **OK** and Performance Center connects to the new Performance Center Server.

Starting, Stopping, and Pinging the Current Server

You can interact with your current server using the [Start](#), [Stop](#), and [Ping](#) commands. These commands appear separately on the Monitor menu.

Starting the Server

- Click **Monitor > Start** and the server is started.

Stopping the Server

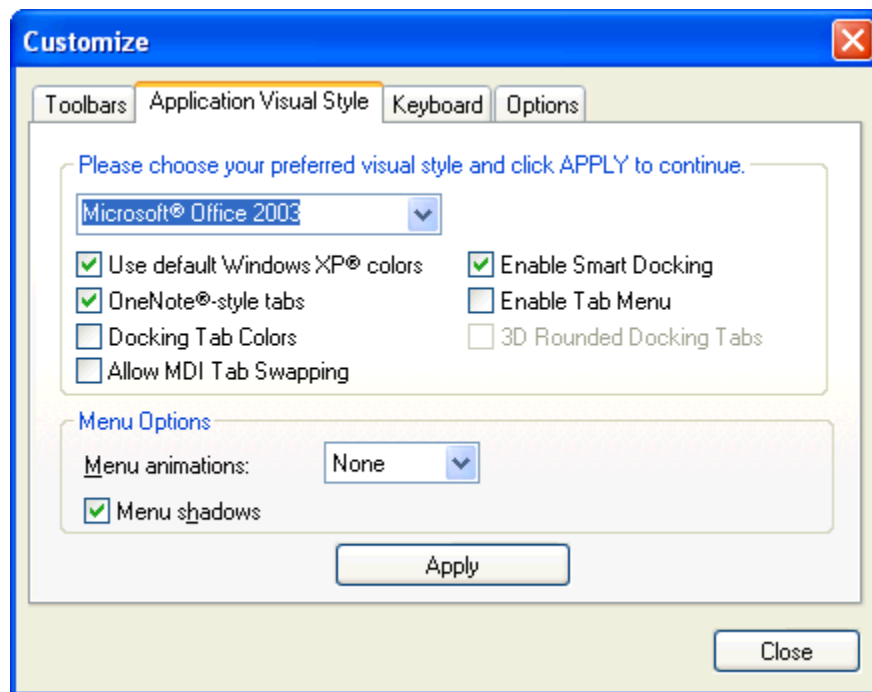
- Click **Monitor > Stop** and the current server is stopped immediately.

Pinging the Server

- Click **Monitor > Ping**. The target Performance Center Server is pinged. If the ping is successful, Performance Center tries to connect to the Server and then opens a Test Result message box with the response from the server.

Customizing the Look and Feel of Performance Center

The **Customize** command of the **Monitor** menu provides access to the **Customize** dialog, which enables you to modify various aspects of Performance Center's look and feel.



Select **Monitor > Customize**, and then choose the appropriate tab for the aspect of Performance Center you want to customize:

- Choose **Toolbars** and select or de-select the Toolbar items you want to appear on the screen. The **Reset** and **Reset All** buttons enable you to change your selections back to the default.
- Choose **Application Visual Style** to modify the look based on common system interfaces. For example, choosing **Microsoft Windows XP** updates the look and feel of the interface to match the default settings of Microsoft Windows XP settings. Other common parameters, based on the system selection, can be modified as well that affect the behavior of windows, dialogs, and mouse control in Performance Center.
- Choose **Keyboard** to set and modify keyboard short cut commands. Use the **Category** drop down menu to locate commands by menu, and then provide a short cut key in the **Press New Shortcut Key** field provided.
- Choose **Options** to modify the way screen tips and short cut keys are displayed on the toolbar and in menus, respectively.

Administrator Login/Administrator Logout

The Administrator Login/Administrator Logout command lets Administrators log in and log out from the Performance Center Server without preventing other users from viewing monitored datasources. During the Performance Center Server installation, the user who installs the Performance Center Repository is automatically granted the role, PERFCNTR_ADMIN, which grants administrative privileges. When you log on as an Administrator, Performance Center authenticates you with the Repository, where you can perform maintenance or enable and disable datasources. It is not necessary to log in to an Performance Center Server to use the Performance Center Client. However, to perform maintenance on monitored datasources, including registering datasources, you must log in as a user with administrative privileges.

TIP: To grant administrative privileges to additional users, you should add users to this role. You can use Embarcadero DBArtisan for this.

Opening the Administrator Login Dialog

- 1 On the **Monitor** menu, click **Administrator Login**.

Performance Center opens the Administrator Login dialog.



- 2 Do one of the following:
 - To use Performance Center without logging on to an Performance Center Server, click **Cancel**.
 - To login to an Performance Center Server, type the user name and password.
- 3 To automatically login the next time you select this Performance Center Server, select **Auto Login**.
- 4 To save and encrypt the credentials for the server locally, select **Auto switch servers**.
- 5 If you do not want Performance Center to display the **Administrator Login** dialog at startup, clear **Show this window at startup** and click **OK**.

TIP: You can also set this in the [Options Editor](#).

Exiting Performance Center

- Click **Monitor > Exit**.

TIP: For information on configuring the Confirmation on Exit message, see Options Editor, [Setting General Options](#).

Datasource Menu

The Datasource provides commands that let you interact with datasources. On the Datasource menu you can manage your datasources and access several dialogs where you can view and modify your datasource Health Indexes and properties. In the Datasource menu you can:

- Profile
- Group
- [Open and close SQL files in the ISQL Window](#)
- [Select datasources](#)
- [Manage datasources](#)
- [Disable/enable datasource monitoring](#)
- [Register datasources](#)
- [Clone datasources](#)
- [Unregister datasources](#)
- Change Group
- Add Datasource to Profile
- Remove Datasource from Profile
- [View and modify datasource thresholds](#)
- [View and modify datasource Health Index properties](#)
- [View and modify datasource recorded statistics](#)
- [View and modify datasource properties](#)
- [Organize datasources](#)

Datasource Alert Maintenance

In the Alert command you can open a dialog where you edit thresholds for individual datasources. On the Datasource menu, you can open the Datasource Alerts Maintenance dialog where you can view alerts for the current datasource. If you are logged into the datasource as an [Administrator](#), you can add, modify, and delete alerts.

Datasource Alert Maintenance

Datasource: EXT0VMCM601 Category: ALL Statistic: ALL

Name	Statistic	Condition	Severity	Enabled
Latch Waits	Latch Waits	> 25	Low	No
Adhoc SQL Hit Ratio	Adhoc SQL Hit Ratio	< 80	Low	No
I/O Error Rate	I/O Error Rate	> 2	High	No
I/O Error Rate	I/O Error Rate	> 1	Low	No
Log Flushes	Log Flushes	> 25	Low	No
Packet Errors	Packet Errors	> 0	Low	No
Log Flush Waits	Log Flush Waits	> 10	Low	No
Log Flush Waits	Log Flush Waits	> 20	High	No
Buffer Cache Hit Ratio	Buffer Cache Hit Ratio	< 95	Low	No
SQL Recompilations	SQL Recompilations	> 5	Low	No
I/O Errors	I/O Errors	> 0	Low	No
Failed Auto-Param Attempts	Failed Auto-Param Attempts	> 0	Low	No
Users Blocked	Users Blocked	> 0	Low	No
Procedure Plan Hit Ratio	Procedure Plan Hit Ratio	< 90	Low	No
Databases Needing Backup	Databases Needing Backup	> 0	Critical	Yes
Databases Low on Space	Databases Low on Space	> 0	Critical	Yes
Buffer Cache Free	Buffer Cache Free	< 1	High	Yes
Failed Auto-Param Attempts	Failed Auto-Param Attempts	> 3	Critical	Yes
Failed Auto-Param Attempts	Failed Auto-Param Attempts	> 1	High	Yes

Buttons: Add... Edit... Delete Attach Notif... Select All Unselect All Close Help

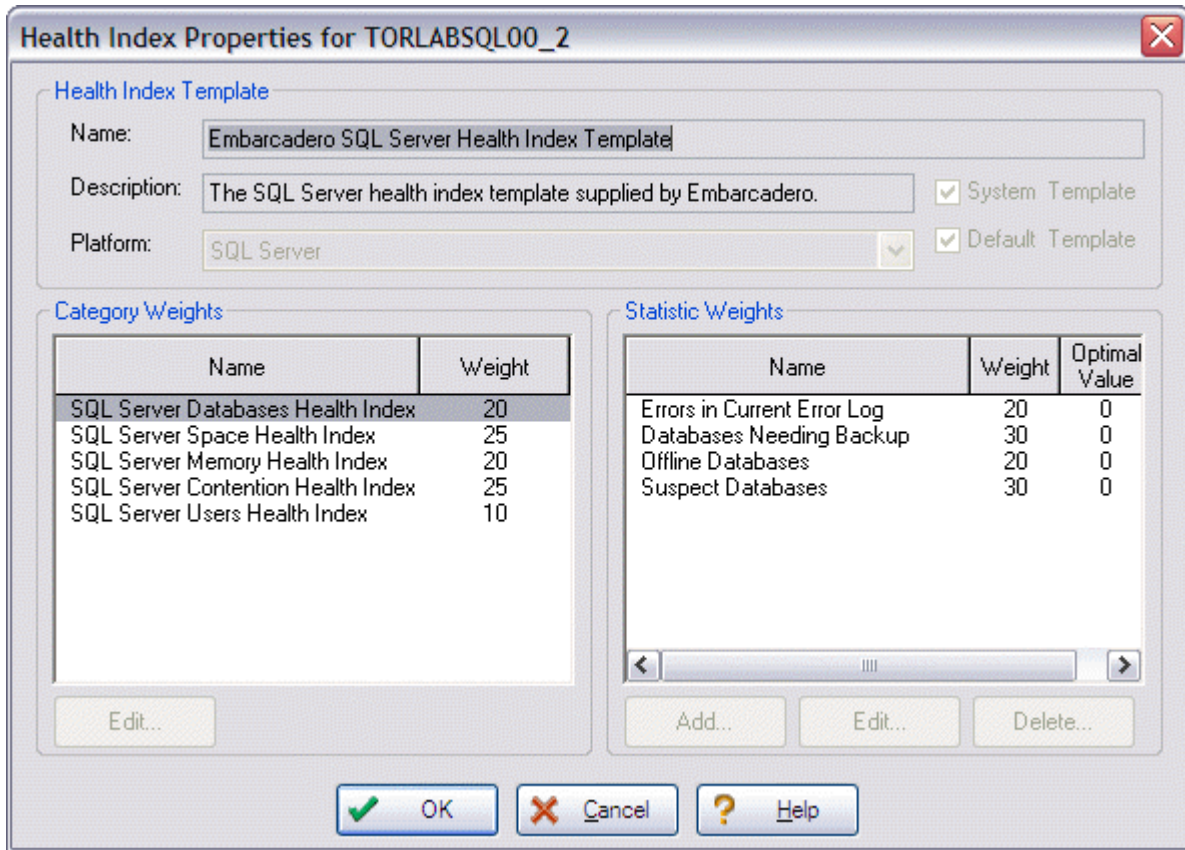
For information on maintaining thresholds, see:

[Alert Templates Maintenance](#)

[Datasource Alerts](#)

Datasource Health Index Properties

In the Health Index Properties dialog you can set the parameters for each monitored datasource Health Index.



Opening the Health Index Properties Dialog

- 1 Click **Main > Datasource**, and then click the target datasource.
- 2 Click **Datasource > Health Index**. The Health Index Properties dialog appears.

For more information, see:

[Health Index Templates](#)

[Using the Health Index Properties Dialog](#)

Health Index Properties Defaults

Performance Center comes with default weighting for categories and statistics. You can use the [Health Index Properties](#) dialog to change the weights to suit your specific needs. The table below describes the Category Weight defaults:

Category	Oracle	SQL Server	Sybase
Contention Health Index	15	25	20
Databases Health Index	n/a	20	10
I/O Health Index	n/a	n/a	10
Memory Health Index	25	20	20
Network Health Index	n/a	n/a	10

Category	Oracle	SQL Server	Sybase
Objects Health Index	15	n/a	n/a
Space Health Index	15	n/a	30
Users Health Index	30	10	n/a

The tables below describes the Statistic Weight defaults for:

- [Oracle](#)
- [SQL Server](#)
- [Sybase](#)

Oracle Statistic Weight Defaults

Category	Statistic	Weight	Optimum Value
Objects Health Index	Objects With Space Deficits	40	0
	Invalid Objects Percentage	5	0
	Tables and Indexes in the Same Tablespace	20	0
	Chained Row Tables	30	0
Users Health Index	Sessions Ratio	15	80
	Users Blocked	25	0
	Users Waiting	10	5
	Number of Users with SYSTEM Tablespace	10	0
	Transaction Ratio	20	80
	Lock Ratio	20	75
Space Health Index	Tablespace Free Chunk Count	20	0
	Tablespace Free Space Percentage	80	0
Contention Health Index	Free Lists Waits	10	0
	Parallel Query Busy Ratio	10	75
	Rollback Contention Ratio	10	1
	Buffer Busy Wait Ratio	20	1
	Users Blocked	30	0
	Latch Miss Ratio	10	0
	Redo Log Space Wait Time	10	5
Memory Health Index	Percentage of Shared Pool Free	10	15
	Memory Sort Ratio	20	95
	Dictionary Cache Hit Ratio	20	90
	Library Cache Hit Ratio	20	95
	Buffer Cache Hit Ratio	30	90

SQL Server Statistic Weight Default

Category	Statistic	Weight	Optimum Value
Databases Health Index	Offline Databases	20	0
	Suspect Databases	30	0
	Errors in Current Error Log	20	0
	Databases Needing Backup	30	0
Space Health Index	Transaction Logs Low on Space	60	0
	Databases Low on Space	40	0
Memory Health Index	Log Cache Hit Ratio	25	90
	Procedure Plan Hit Ratio	25	90

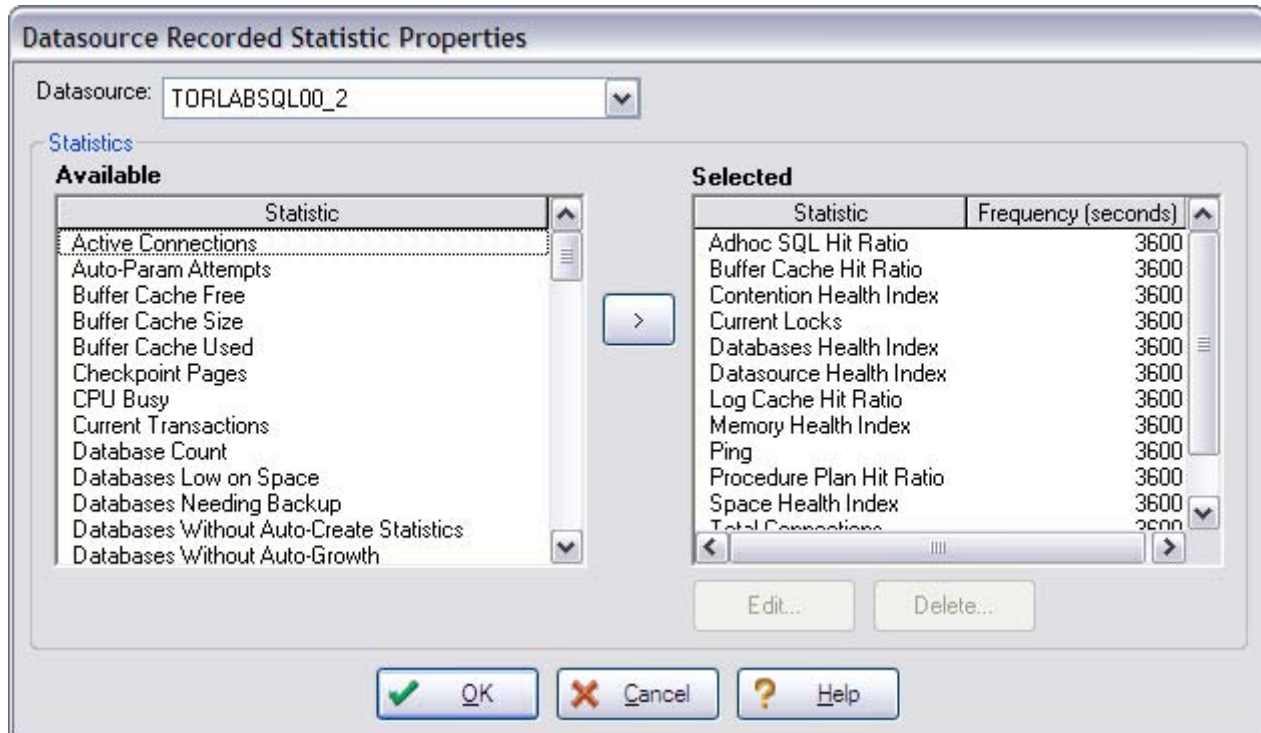
Category	Statistic	Weight	Optimum Value
	Adhoc SQL Hit Ratio	15	80
	Buffer Cache Hit Ratio	35	90
Contention Health Index	I/O Busy	20	75
	CPU Busy	25	75
	Latch Waits	10	0
	Deadlocks	15	0
	Users Blocked	30	0
Users Health Index	Failed Auto-Param Attempts	30	0
	SQL Recompilations	70	10

Sybase Statistic Weight Default

Category	Statistic	Weight	Optimum Value
Space Health Index	Transaction Logs Low on Space	50	10
	Databases Low on Space	50	10
Databases Health Index	Users Blocked	30	0
	Databases With Suspect Pages	30	0
	Suspect Databases	40	0
I/O Health Index	I/O Error Rate	30	1
	I/O Busy Rate	30	95
	Delayed I/O's	30	10
	Outstanding I/O's	10	10
Memory Health Index	Large I/O Denied	20	0
	Clean Buffer Grab Rate	20	90
	Procedure Cache Hit Rate	30	95
	Data Cache Hit Rate	30	95
Contention Health Index	I/O Operating System Limit Delays	5	10
	I/O Engine Configuration Limit Delays	5	10
	I/O Server Configuration Limit Delays	5	10
	Disk I/O Structure Delays	5	10
	Device Waits	10	5
	Deadlock Rate	30	1
Network Health Index	Blocking Lock Rate	40	5
	Packet Errors	40	5
	Network Delays	60	5

Datasource Recorded Statistics

In the Datasource Recorded Statistics dialog you can select statistics that you want Performance Center to write to the Repository.



Opening the Datasource Recorded Statistics Dialog

- 1 On the **Main** toolbar, click the **Datasource** list, and then click the target datasource.
- 2 On the **Datasource** menu, click **Recorded Statistics**.

Performance Center opens the Datasource Recorded Statistics dialog.

For more information, see:

Recorded Statistics

[Using the Datasource Recorded Statistics Dialog](#)

Edit Menu

The Edit menu provides commands that let you work with SQL scripts in the ISQL Window. The Edit menu is only available when an ISQL Window is the active window in the Workbook. The table below describes the commands available on the Edit menu:

Command	Description
Undo	Undoes the last text operation in the ISQL window.
Redo	Redoes the last text operation in the ISQL window.
Cut	Cuts selected text from the ISQL window and places on the Windows clipboard.
Copy	Copies text from the ISQL window and places it in the Windows clipboard.
Paste	Pastes text from the Windows clipboard into the ISQL window.
Delete	Deletes selected text in the ISQL window.

Command	Description
Select All	Selects all text in the ISQL window.
Indent	Moves the selected text to the right.
Outdent	Moves the selected text to the left.
Find	Opens the Find dialog where you can specify a search string.
Repeat Find	Opens the Find dialog where you can continue searching for the target search string.
Replace	Opens the Replace dialog where you can specify a search string and replacement string.
Goto	Opens the Go To dialog where you can specify a line number to jump to in the ISQL window.
Toggle Bookmark	Toggles on/on a bookmark on the current line in the ISQL window.
Next Bookmark	Moves the cursor to the next line with a bookmark.
Previous Bookmark	Moves the cursor to the previous line with a bookmark.
Clear All Bookmarks	Removes all bookmarks from the ISQL window.

For detailed information on using the ISQL Window, see [SQL Scripting](#)

Maintenance Menu

The Maintenance menu provides commands that let you access the Performance Center Maintenance dialogs. In the Maintenance menu you can open the following dialogs:

- [Address Maintenance](#)
- [Notification Maintenance](#)
- [Action Maintenance](#)
- [Schedule Maintenance](#)
- [Report Maintenance](#)
- [Data Extraction Job Maintenance](#)
- [Statistic Maintenance](#)
- [Alert Template Maintenance](#)
- [Health Index Template Maintenance](#)
- [Recorded Statistic Maintenance](#)

Report Menu

In the Report menu you can open the Select Historical Report dialog, which includes several types of reports that you can run against your registered datasources. On the Report menu you can also input or update the Statspack login for an Oracle datasource.

Performance Center uses the Microsoft Task Scheduler component of IE for scheduling some reports. You must have the start the Scheduler service to schedule Performance Center reports.

The table below describes the types of reports available in Performance Center:

Report Name	Description
Average User Database Volume	Illustrates the connection traffic for a database over time. Using the historically maintained amounts of connection volume, this report presents to middle management and staff an at-a-glance overview of the database traffic volume.
Database Downtime Detail	Illustrates the percentage of uptime a database has experienced for a given period of time.
Database Performance Comparison	Compares the overall performance index of various databases to user-defined points in time.
Enterprise Connection Database Volume	Illustrates the comparison of user traffic across all monitored databases in an enterprise. Using the historical obtained amounts for user traffic, this report presents to senior management an at-a-glance overview of how their databases compare to one another in user volume.
Enterprise Database Availability	Illustrates the uptime of critical databases. Because Performance Center polls its monitored datasources on a continuous basis, it calculates the availability of the databases it tracks. This report will present to senior management an at-a-glance overview of how enterprise databases look in terms of availability.
Enterprise Database Performance	Illustrates the overall performance of all databases in an enterprise. Using the historically maintained global performance index for each database, this report aggregates all monitored databases together to come up with an overview appropriate for senior management.
Enterprise Database Performance by Category	Illustrates the overall performance of all databases in an enterprise by performance category. Using the historically maintained building block indexes for each database, this report aggregates all monitored databases together to come up with an overview appropriate for senior and middle management.
Multiple Statistic Baseline Analysis	Reports on multiple statistics for one datasource over a period of time and compare these values to the same statistics over a different time period.
Multiple Statistic Trend Analysis	Reports on multiple statistics for one datasource.
Oracle Buffer Wait Statistics	Reports on the top Buffer Wait statistics over various sampled snapshots.
Oracle Buffer Wait Statistics Comparison to Baseline	Generates two Buffer Wait Statistics reports and then compares them.
Oracle Instance Efficiency Ratios	Reports on the following statistics: Buffer Cache Hit Ratio, Dictionary Cache Hit Ratio, Library Cache Hit Ratio, Memory Sort Ratio, Latch Miss Ratio, and Rollback Contention Ratio.
Oracle Instance Efficiency Ratios Comparison to Baseline	Reports on the following statistics: Buffer Cache Hit Ratio, Dictionary Cache Hit Ratio, Library Cache Hit Ratio, Memory Sort Ratio, Latch Miss Ratio, and Rollback Contention Ratio and compares their values against baseline values.
Oracle Load Profile	Reports on the following statistics: Redo Log Size, Logical Reads, Logical Changes, Physical Reads, Physical Writes, Disk Sorts, Max Connections, and Max Transactions.
Oracle Load Profile Comparison to Baseline	Reports on the following statistics: Redo Log Size, Logical Reads, Logical Changes, Physical Reads, Physical Writes, Disk Sorts, Max Connections, and Max Transactions and compares their values against baseline values.
Oracle Top Wait Events	Reports on the top Oracle wait events. This report requires installation of the Oracle Statspack on the monitored datasource.
Oracle Top Wait Events Comparison to Baseline	Reports on the top Oracle wait events. This report requires installation of the Oracle Statspack on the monitored datasource and compares their values against baseline values.
SGA Regions and Memory Areas	Reports important Oracle startup parameters.

Report Name	Description
Statistical Trend Analysis	Compares statistics for user-specified time periods. This report presents to middle management and staff various comparisons of defined statistics.
Worst Databases	Displays the worst performing databases (as indicated by performance indexes) which exist in an enterprise. Using the historically maintained global performance index for each database, this report averages the index for the user-specified time interval and shows the top 'x' worst databases.

For more information, see [Reports](#).

View Menu

The View menu provides commands that let you customize the display of the Performance Center Client UI. The View menu is a dynamic drop-down menu that changes depending on your current view. For example, when you are in the Home view, the View menu lists all Performance Center statistics for the current datasource. In the View menu you can:

- [Customize the display](#)
- [Open the Server Manager](#)
- [Open the Event Log](#)
- [Access target datasource views](#)

Customizing the Display

On the View menu, you can toggle the display of the following:

- [Explorer](#)
- [Output Window](#)
- [Enterprise View](#)
- [Workbook](#)
- [Toolbar](#)
- [Status Bar](#)
- [Full Screen Mode](#)

Opening the Server Manager

The Server Manager command opens the Server Manager, which is a series of pages that display detailed information about the Performance Center Server(s). You can perform maintenance on the Performance Center Server through the Server Manager. It is also an alternate way of accessing the Performance Center Server outside of the Performance Center Client.

NOTE: You must be logged into the datasource as a user with [Administrator](#) privileges to access this feature.

Opening the Server Manager

- On the **View** menu, click **Server Manager**.

Performance Center opens the Performance Center Server Manager.

NOTE: The Server Manager is also accessible from the Performance Center Server machine.

Full Screen Mode

The Full Screen command provides full screen mode capabilities, which let you conceal the application framework and use the entire monitor area. Full screen mode hides any other applications running on the computer and uses every available pixel for the application.

Activating Full Screen Mode

- On the **View** menu, click **Full Screen**.

Performance Center expands the application to fit the entire monitor area.

NOTE: The Full Screen Mode icon is a stand-alone floating toolbar.

Dismissing Full Screen Mode

Performance Center offers two ways to dismiss full screen mode:

- To restore the application to the default size, click **Close Full Screen**.

Datasource Views

The View menu displays a list of all views for the current datasource. You can also access all views from the [Explorer](#).

Window Menu

The Window menu provides commands where you can customize the display of open windows in Performance Center. In the Window menu you can:

- Cascade and tile open windows
- Arrange icons
- Toggle between open windows
- Minimize, restore, and close windows

Help Menu

The Help menu provides commands that you can:

- [Open Performance Center Help](#)
- [Connect to the Embarcadero Technologies Web site](#)
- [View Performance Center version and licensing information](#)

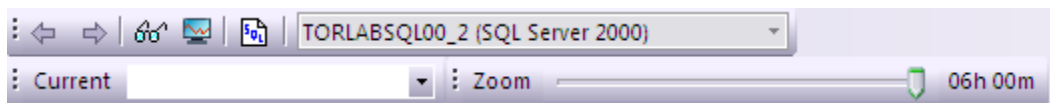
Shortcut Menus

Performance Center includes many shortcut menus that let you access the most frequently used commands for each component of the application. To view any shortcut menu, right-click anywhere on the Performance Center workspace. The shortcut menus list commands that pertain to the selected item or object. Several of the shortcut menu commands have icons next to them that let you associate the command with the image.





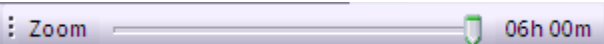

The Expert Guides are available through the shortcut menu. While in any Performance Center view, right-click any statistic, vital sign, or category, and then click Help to open the [Oracle Expert Guide](#), [SQL Server Expert Guide](#), [Sybase Expert Guide](#), or the [DB2 Expert Guide](#).

Toolbar

The Performance Center toolbar includes several icons and a list that let you access features of the application.



The table below describes the icons on the Performance Center toolbar:

Icon	Command	Description
	Back	Opens the previous view window.
	Next	Opens the proceeding view window.
	Monitor Events	Opens the Event Log .
	Select	Opens the Select Performance Center Server dialog .
	Zoom	Choose a percentage value by clicking and dragging the slider. This represents the percentage amount of the server cache to display when viewing graphs.
	Current	Specifies the date and time. Click the drop down menu to access a calendar view. You can select a date and click Set to change it.

The list includes all [registered datasources](#) available on the current server. To change datasources, click the list, and then click the target datasource.

Maintenance Dialog

Performance Center includes several maintenance dialogs designed to help you monitor your enterprise:

- [Address Maintenance](#)

- [Notification Maintenance](#)
- [Action Maintenance](#)
- [Statistic Maintenance](#)
- [Schedule Maintenance](#)
- [Report Maintenance](#)
- [Alert Template Maintenance](#)
- [Health Index Template Maintenance](#)
- [Recorded Statistic Template Maintenance](#)

For the Alert and Health Index Template Maintenance dialogs, Performance Center includes standards, which were created to help you determine what are industry recommendations for each performance category statistic. You can use these dialogs to customize thresholds so that you can detect and pinpoint problem areas in any monitored datasource.

Address Maintenance

The Address Maintenance dialog stores information about who gets notified and by what means. It is a generic, reusable list of contacts and the ways you can communicate with them. On the Maintenance menu, you can open the Address Maintenance dialog where you can view addresses. If you are logged into the datasource as an [Administrator](#), you can add, modify, and delete addresses.

For information on maintaining addresses, see [Addresses](#).

Notification Maintenance

The Notifications Maintenance dialog stores reusable notifications. You can use this dialog to define a notification and select the [addresses](#) to send the notification. For example, you could create a notification called Database Down, which includes the addresses of the DBAs you want to page when a database goes down. On the Maintenance menu, you can open the Notification Maintenance dialog where you can view notifications. If you are logged into the datasource as an [Administrator](#), you can add, modify, and delete notifications.

For information on setting notifications, see [Notifications](#).

Action Maintenance

The Action Maintenance dialog stores sample and user-defined actions, which Performance Center launches in response to a [threshold](#) violation. For example, you can define an action in Performance Center that will launch a script, if a database goes down, to bring the database back up. On the Maintenance menu, you can open the Action Maintenance dialog where you can view actions. If you are logged into the datasource as an [Administrator](#), you can add, modify, and delete actions.

For information on maintaining actions, see [Actions](#).

Sample Actions

Performance Center includes several sample actions to help you use the application.

The sample actions are provided to illustrate some of the many ways in which you can obtain custom information when a threshold is violated. In these examples, we generate HTML reports that can be viewed to obtain additional information surrounding a problem. While these are robust and somewhat involved, remember that an Action can be something as simple as a CMD file if you so desire. The table below describes the components of the sample actions provided by Embarcadero Technologies:

Sample Action	Description
CSCRIPT.EXE	In the command field of an action is the executable: CSCRIPT.EXE. This is a Windows script interpreter that comes with the Windows Scripting Host. It is used to interpret the wsf file provided in the Arguments field.
ACTION.WSF	This is a Windows Scripting File (wsf) that is the primary driver in building the report. All wsf files are interpreted by either cscript.exe or wscript.exe, both of which are provided with the Windows Scripting Host. ACTION.WSF takes the following parameters: /c - Connection string /s - Subject for the e-mail /m - E-mail message text /q - The name of the file containing the query /r - E-mail recipient

NOTE: You can view and edit ACTION.WSF. It is in the following directory: %install_directory%\scripts\actions.

Within ACTION.WSF you will find a reference to ACTION.HTM; also located in the Actions directory where it serves as the template for the HTML reports.

TIP: To learn more about scripting files, visit <http://msdn.microsoft.com/scripting>.

The last file of significance in the sample actions is ADORRecordset.xsl. This extensible style sheet is used to transform XML data into an HTML table. ADORRecordset.xsl is referenced in the ConvertRecordsetToHTML function in ACTION.WSF.

You can use this information to learn how to design custom reports using Actions, or edit any of the above components to create the types of reports you need for your particular business.

The table below lists the sample actions included with Performance Center:

Name	Description
Oracle < = 8.0.6: TS Low on Space	E-mails a list of tablespaces that have less than 10% free space.
Oracle 8i: TS Low on Space	E-mails a list of tablespaces that have less than 10% free space.
Oracle: Blocking Locks	E-mails a list of blocking locks.
Oracle: Buffer Busy Waits	E-mails a list of buffer busy waits.
Oracle: Chained Tables	E-mails a list of chained tables.
Oracle: Currently Running SQL	E-mails a list of currently running SQL.
Oracle: Data Dictionary Cache Details	E-mails the data dictionary cache details.
Oracle: Disk Sort Detail	E-mails the disk sort detail.
Oracle: Invalid Objects	E-mails a list of invalid objects.
Oracle: Kill Blocked Processes	Kills blocked processes.
Oracle: Kill Blocking Processes	Kills blocking processes.

Name	Description
Oracle: Latch Waits	E-mails a list of latch waits.
Oracle: Library Cache Usage	E-mails the library cache usage.
Oracle: Objects With Extent Problems	E-mails a list of objects with extent problems.
Oracle: Objects With Space Deficits	E-mails a list of objects that cannot extend.
Oracle: Open Cursors	E-mails a list of open cursors.
Oracle: Prolonged Lock Waits	E-mails a list of prolonged lock waits.
Oracle: Recompile Invalid Objects	Recompiles all invalid objects.
Oracle: Session Waits	E-mails a list of session waits.
Oracle: Shared Pool Contents	E-mails a list of the contents of the shared pool.
Oracle: SQL in Shared Area	E-mails a list of SQL in the shared area.
SQL Server 7.x: Databases Low on Space	E-mails a list of databases that are low on space.
SQL Server 8.x: Databases Low on Space	E-mails a list of databases that are low on space.
SQL Server: Blocking Locks	E-mails a list of blocking locks.
SQL Server: Buffer Cache Usage	E-mails the current buffer cache usage.
SQL Server: Databases Needing Backup	E-mails a list of databases needing backup.
SQL Server: Kill Blocked Processes	Kills blocked processes.
SQL Server: Kill Blocking Processes	Kills blocking processes.
SQL Server: Processes	E-mails a list of database processes.
SQL Server: Today's Errors	E-mails a list of errors from the SQL Server error log that occurred the current day.
SQL Server: Transaction Logs Low on Space	E-mails a list of transaction logs that are low on space.
Sybase: Blocking Locks	E-mails a list of blocking locks.
Sybase: Data Cache Activity	E-mails data cache activity detail.
Sybase: Processes	E-mails a list of database processes.

Statistic Maintenance

The Statistic Maintenance dialog displays the properties of tracked statistics. You can use this dialog to [manipulate thresholds](#) for the available statistics. On the Maintenance menu, you can open the Statistic Maintenance dialog where you can monitor all statistics for all of your datasources.

For information on maintaining statistics, see [Statistics](#).

Schedule Maintenance

In the Schedule Maintenance dialog you can create a time interval where Performance Center will not send notifications about any normal threshold violations or performance problems. For example, if you know you will run an Oracle database cold backup every day between three and four am, you can use the Schedule Maintenance to create a blackout schedule so that you are not notified.

On the Maintenance menu, you can open the Schedule Maintenance dialog where you can view schedules. If you are logged into the datasource as an [Administrator](#), you can add, edit, delete, and attach schedules.

For information on maintaining schedules, see:

[Schedules](#)

[Notifications](#)

[Enterprise Alerts](#)

Report Maintenance

You can generate reports and write them to any location. On the Maintenance menu, you can open the Report Maintenance dialog where you configure all reports for all of your datasources.

NOTE: Performance Center uses the Microsoft Task Scheduler component of IE for scheduling some reports. You must start the Scheduler service to schedule Performance Center reports.

For information on maintaining reports, see [Reports](#).

Data Extraction Jobs

The Data Extraction Jobs maintenance dialog is command central of the data extraction jobs feature. It's here that you add, edit, or run jobs to export data to tables you have already created or new ones to see statistics pulled from datasources you specify. You can move data across platforms to make seeing the data easier for users.

For more information on scheduled reports, see [Data Extraction Jobs](#).

Alert Template Maintenance

Alert Templates are logical containers used to define and deploy thresholds to monitored datasources. Embarcadero supplies a base template to serve as a starting point for each new template creation effort. Once you create a template and store it in the Performance Center Repository, you can then apply the template along with a notification setup/schedule to any monitored datasource. You can use the Alert Template Maintenance dialog to edit thresholds on a global level as well as for individual datasources. You can delete templates from the repository when you no longer need them.

On the Maintenance menu, you can open the Alert Templates dialog where you can view Alert Templates. If you are logged into the datasource as an [Administrator](#), you can add, modify, clone, apply, and delete Alert Templates.

For information on maintaining Alert Templates, see [Alert Templates](#).

Embarcadero provides a default template for each supported database platform:

- [Oracle](#)
- [SQL Server](#)
- [Sybase](#)
- [DB2](#)

Embarcadero Oracle Alert Template

Name	Condition	Severity	Enabled
Buffer Busy Wait Ratio	greater than or equal to 2	Critical	Yes
	greater than or equal to 1	High	Yes
	greater than or equal to 0.5	Low	No
Buffer Cache Hit Ratio	less than 80	Critical	Yes
	less than 90	High	Yes
	less than 95	Low	No
Dictionary Cache Hit Ratio	less than 80	Critical	Yes
	less than 90	High	Yes
	less than 98	Low	No
Free List Waits	greater than 10	Critical	Yes
	greater than 5	High	Yes
	greater than 0	Low	No
Latch Immediate Miss Rate	greater than 2	Critical	Yes
	greater than or greater than 1	High	Yes
	greater than or equal to 0.5	Low	No
Latch Miss Ratio	greater than 2	Critical	Yes
	greater than or greater than 1	High	Yes
	greater than or equal to 0.5	Low	No
Latch Sleep Ratio	greater than 2	Critical	Yes
	greater than or equal to 1	High	Yes
	greater than or equal to 0.5	Low	No
Library Cache Hit Ratio	less than 80	Critical	Yes
	less than 85	High	Yes
	less than 90	Low	No
Memory Sort Ratio	less than 80	Critical	Yes
	less than 90	High	Yes
	less than 95	Low	No
Extent Deficits (Objects With Space Deficits)	greater than 0	Critical	Yes
Parallel Query Busy Ratio	equal to 100	Critical	Yes
	greater than 90	High	Yes
	greater than 80	Low	No
% Free Memory in Shared Pool (Percentage of Shared Pool Free)	less than 5	Critical	Yes
	less than 7	High	Yes
	less than 10	Low	No
Datasource Down (Ping)	equal to 2	Critical	Yes

Name	Condition	Severity	Enabled
Listener Down (Ping)	equal to 3	Critical	Yes
Redo Log Space Requests	greater than 5	Critical	Yes
	greater than 2	Moderate	Yes
	greater than 0	Low	No
Redo Log Wait Time (Red Log Space Wait Time)	greater than 200	Critical	Yes
	greater than 100	High	Yes
	greater than 49	Low	No
Rollback Segment Wait Ratio (Rollback Contention Ratio)	greater than 2	Critical	Yes
	greater than or equal to 1	High	Yes
	greater than or equal to 0.5	Low	No
Users Blocked	greater than 10	Critical	Yes
	greater than 5	High	Yes
	greater than 0	Low	No
Users Waiting	greater than 10	Critical	Yes
	greater than 5	High	Yes
	greater than 0	Low	No

Embarcadero SQL Server Alert Template

Name	Condition	Severity	Enabled
Adhoc SQL Hit Ratio	less than 60	Critical	Yes
	less than 70	High	High
	less than 80	Low	Yes
Buffer Cache Free	less than 2	Low	Yes
	less than 1	High	High
	less than 0.5	Critical	Yes
Buffer Cache Hit Ratio	less than 80	Critical	Yes
	less than 90	High	Yes
	less than 95	Low	No
Databases Low on Space	greater than 0	Critical	Yes
Databases Needing Backup	greater than 0	Critical	Yes
Deadlocks	greater than 0	Critical	Yes
Failed Auto-Param Attempts	greater than 0	Low	No
	greater than 1	High	Yes
	greater than 3	Critical	Yes
I/O Error Rate	greater than 5	Critical	Yes
	greater than 2	High	No
	greater than 1	Low	No
I/O Errors	greater than 0	Low	No
	greater than 1	High	Yes
	greater than 3	Critical	Yes
Latch Waits	greater than 25	Low	Yes
	greater than 50	High	Yes
	greater than 75	Critical	Yes
Log Cache Hit Ratio	less than 90	Low	Yes
	less than 60	Critical	Yes
	less than 70	High	Yes
Log Flush Waits	greater than 20	High	No
	greater than 10	Low	No
	greater than 30	Critical	Yes
Log Flushes	greater than 25	Low	Yes
	greater than 50	High	Yes
	greater than 100	Critical	Yes
Offline Databases	greater than 0	Critical	Yes
Packet Errors	greater than 0	Low	No
	greater than 1	High	Yes

Name	Condition	Severity	Enabled
	greater than 3	Critical	Yes
Datasource Down (Ping)	is equal to 2	Critical	Yes
Procedure Plan Hit Ratio	less than 90	Low	No
	less than 60	Critical	Yes
	less than 70	High	Yes
SQL Recompilations	greater than 10	High	No
	greater than 5	Low	Yes
	greater than 20	Critical	Yes
Suspect Databases	greater than 0	Critical	Yes
Transaction Logs Low on Space	greater than 0	Critical	Yes
Users Blocked	greater than 10	Critical	Yes
	greater than 5	High	Yes
	greater than 0	Low	No

Embarcadero Sybase Alert Template

Name	Condition	Severity	Enabled
Blocking Lock Rate	greater than 10	High	Yes
	greater than 15	Critical	No
	greater than 3	Low	No
Clean Buffer Grab Rate	less than 90	Critical	Yes
Data Cache Hit Rate	less than 80	Critical	Yes
	less than 90	High	Yes
	less than 95	Low	No
Deadlock Rate	greater than 5	Critical	Yes
	greater than 2	High	Yes
Deadlocks	greater than 1	Critical	No
Device Waits	greater than 1	Critical	No
Disk I/O Structure Delays	greater than 40	Critical	No
	greater than 20	High	No
	greater than 10	Low	No
I/O Engine Configuration Limit Delays	greater than 40	Critical	Critical
	greater than 10	Low	Low
	greater than 20	High	High
I/O Error Rate	greater than 5	Critical	Yes
	greater than 2	High	No
	greater than 1	Low	No
I/O Operating System Limit Delays	greater than 40	Critical	No
	greater than 20	High	No
	greater than 10	Low	No
I/O Server Configuration Limit Delays	greater than 40	Critical	No
	greater than 20	High	No
	greater than 10	Low	No
Large I/O Denied	greater than 30	Critical	Yes
	greater than 10	High	Yes
	greater than 1	Low	No
Lock Promotions	greater than 1	Critical	No
Network Delays	greater than 10	Critical	Yes
	greater than 2	High	No
Datasource Down (Ping)	is equal to 2	Critical	Yes
Procedure Cache Hit Rate	less than 80	Critical	Yes
	less than 90	High	Yes
	less than 95	Low	No

Name	Condition	Severity	Enabled
ULC Flushes	greater than or equal to 20	Critical	No
	greater than 10	High	Yes
	greater than 5	Low	No
Used to Max Locks	greater than 90	Critical	No
	greater than 75	High	No

Embarcadero DB2 Alert Template

Name	Condition	Severity	Enabled
Bufferpool Index Hit Ratio	< 90	Low	No
Deadlocks Detected	>= 1	High	Yes
Deadlocks Detected	> 0	Low	No
DMS Space Utilization	> 85	Critical	Yes
DMS Space Utilization	> 80	High	Yes
DMS Space Utilization	> 75	Low	No
Hash Join Overflow Percentage	> 10	Critical	Yes
Hash Join Overflow Percentage	> 5	High	Yes
Hash Join Overflow Percentage	> 0	Low	No
Inaccessible Containers	> 0	Critical	Yes
Lock Timeouts	>= 3	Critical	Yes
Lock Timeouts	>= 1	High	Yes
Lock Timeouts	> 0	Low	No
Lock List Utilization	> 85	Critical	Yes
Lock List Utilization	> 80	High	Yes
Lock List Utilization	> 75	Low	No
Lock Wait Time	> 1000	Critical	Yes
Lock Wait Time	> 800	High	Yes
Lock Wait Time	> 500	Low	No
Max Connection Utilization	> 85	Critical	Yes
Max Connection Utilization	> 80	High	Yes
Max Connection Utilization	> 75	Low	No
Monitor Heap Utilization	> 85	Critical	Yes
Monitor Heap Utilization	> 80	High	Yes
Monitor Heap Utilization	> 75	Low	No
Abnormal State Tablespaces	> 0	Critical	Yes
Active Log Utilization	> 85	Critical	Yes
Active Log Utilization	> 80	High	Yes
Active Log Utilization	> 75	Low	No
Agent Utilization	> 85	Critical	Yes

Name	Condition	Severity	Enabled
Agent Utilization	> 80	High	Yes
Agent Utilization	> 75	Low	No
Applications Waiting on Locks	>= 5	Critical	Yes
Applications Waiting on Locks	>= 3	High	Yes
Applications Waiting on Locks	> 0	Low	No
Average Lock Wait Time	> 25	Critical	Yes
Average Lock Wait Time	> 20	High	Yes
Average Lock Wait Time	> 10	Low	No
Agents Waiting on Tokens	>= 5	Critical	Yes
Agents Waiting on Tokens	>= 3	High	Yes
Agents Waiting on Tokens	> 0	Low	No
Bufferpool Index Hit Ratio	< 80	Critical	Yes
Bufferpool Index Hit Ratio	< 85	High	Yes
Deadlocks Detected	>= 3	Critical	Yes
Bufferpool Hit Ratio	< 80	Critical	Yes
Bufferpool Hit Ratio	< 85	High	Yes
Bufferpool Hit Ratio	< 90	Low	No
Catalog Cache Hit Ratio	< 80	Critical	Yes
Catalog Cache Hit Ratio	< 85	High	Yes
Catalog Cache Hit Ratio	< 90	Low	No
Catalog Cache Overflows	>= 3	Critical	Yes
Catalog Cache Overflows	>= 1	High	Yes
Catalog Cache Overflows	> 0	Low	No
Database Heap Utilization	> 85	Critical	Yes
Database Heap Utilization	> 80	High	Yes
Database Heap Utilization	> 75	Low	No
Database Status	> 0	Critical	Yes
Package Cache Hit Ratio	< 75	Critical	Yes
Package Cache Hit Ratio	< 80	High	Yes
Package Cache Hit Ratio	< 85	Low	No
Package Cache Overflows	>= 3	Critical	Yes
Package Cache Overflows	>= 1	High	Yes
Package Cache Overflows	> 0	Low	No
Percent of Agents Stolen	> 10	Critical	Yes
Percent of Agents Stolen	> 5	High	Yes
Percent of Agents Stolen	> 0	Low	No
Shared Workspace Hit Ratio	< 75	Critical	Yes
Shared Workspace Hit Ratio	< 80	High	Yes

Name	Condition	Severity	Enabled
Shared Workspace Hit Ratio	< 85	Low	No
SMS Space Utilization	> 85	Critical	Yes
SMS Space Utilization	> 80	High	Yes
SMS Space Utilization	> 75	Low	No
Sort Heap Utilization	> 85	Critical	Yes
Sort Heap Utilization	> 80	High	Yes
Sort Heap Utilization	> 75	Low	No
Sort Overflow Percentage	>= 3	Critical	Yes
Sort Overflow Percentage	>= 1	High	Yes
Sort Overflow Percentage	> 0	Low	No
Tablespaces Low on Space	> 0	Critical	Yes
Database Backup Required	= 0	Critical	Yes
Victim Page Cleans Ratio	> 20	Critical	Yes
Victim Page Cleans Ratio	> 10	High	Yes
Victim Page Cleans Ratio	> 0	Low	No
Physical Read Percentage	> 50	Critical	Yes
Physical Read Percentage	> 30	High	Yes
Physical Read Percentage	> 25	Low	No
Datasource Down	= 2	Critical	Yes

Health Index Template Maintenance

Health Index Templates are the logical containers used to define the various performance indexes used to globally measure database performance. You can define which performance categories (Memory, Space, etc.) to include in the global database performance index, and weight all categories according to their level of importance. Embarcadero Technologies supplies three base templates (one for each supported DBMS) to serve as a starting point for each new template creation effort. Once you create a template and store it in the Performance Center Repository, you can then apply the template along with a notification setup/schedule to any monitored datasource. You can use the Health Index Template Maintenance dialog to edit performance categories and statistics for your enterprise on a global level and for individual datasources. You can delete templates from the repository when you no longer need them.

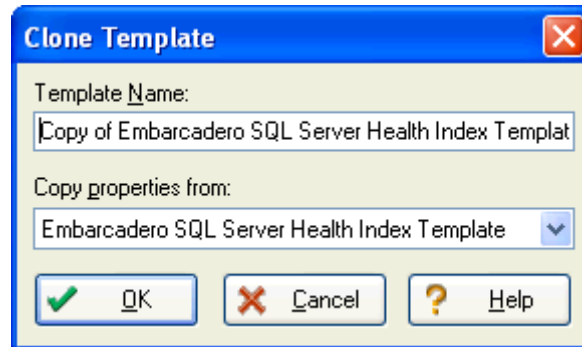
You can view the different indexes and how it is being calculated. When you open the Health Index Template Properties Edit dialog you can see what statics are being used on the right and what the weight of each statistic is. If you want to edit the specific indexes, you need to clone a copy of the default template in the Health Index Templates dialog. You cannot change the default index but you can edit the cloned copy, make it the default and apply it to your datasources.

On the Maintenance menu, you can open the Health Index Templates dialog where you can view Health Index Templates. If you are logged into the datasource as an [Administrator](#), you can add, modify, clone, apply, and delete Health Index Templates.

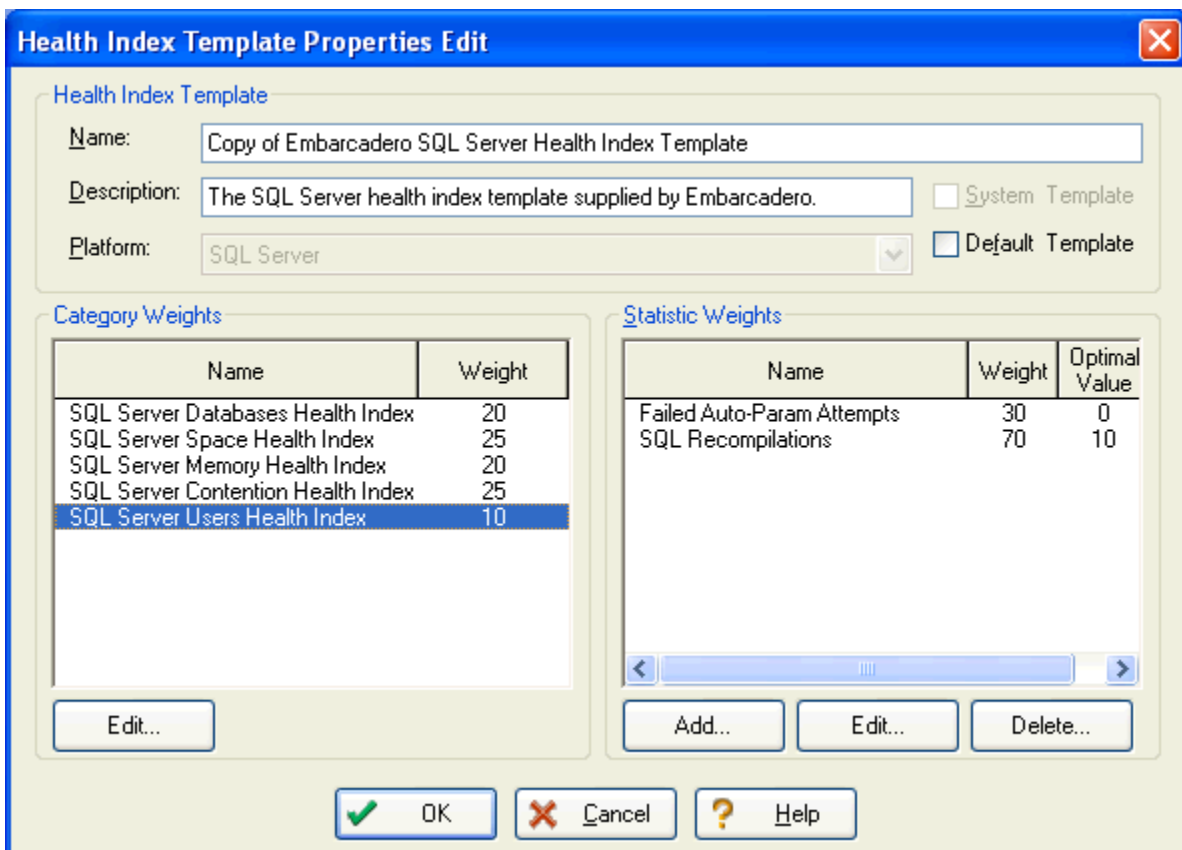
Editing and Calculating the Health Index Templates

In the following steps, we will clone and edit the weights of a Health Index.

- 1 Select **Maintenance -> Health Index Templates** and the Health Index Templates appears.
- 2 Select the Health Index specific for your database plate, click **Clone** and the Clone Template dialog appears. The template name will be appended with “copy of” and will copy the properties from the selected template.



- 3 Click **OK** and the cloned index appears in the list.
- 4 Select the newly created index, click **Edit** and the **Health Index Template Properties Edit** dialog appears.



- 5 You can edit a Category Weight for an index listed. Select an index in the list and click **Edit**. In the Health Index Weight dialog, change the Weight setting and click **OK**.

- 6 When you select an index in the Category Weights panel, the Statistic Weights and Optimal Value for the selected index appears in the right panel. You can then add, edit, or delete the health index data for that index.

NOTE: The total weights must always equal 100 as you make your edits.

- 7 Once you have edited the cloned copy, select the **Default Template** option to make it the default and click **OK**.
- 8 A warning message appears telling you that you must apply the changed health index template to a monitored datasource. Click **OK** and the Select Datasource(s) dialog appears.
- 9 Select a registered datasource from the list and click **OK**. The health Index template now shows that the edited index is now the default.

The tables below lists the Health Index Template defaults for:

- [Oracle](#)
- [SQL Server](#)
- [Sybase](#)
- [DB2](#)

Oracle Health Index Template

Category	Statistic	Weight	Optimum Value
Objects Health Index	Objects With Space Deficits	45	0
	Invalid Objects Percentage	5	0
	Tables and Indexes in the Same Tablespace	20	0
	Chained Row Tables	30	0
Users Health Index	Sessions Ratio	15	80
	Users Blocked	25	0
	Users Waiting	10	5
	Number of Users with SYSTEM Tablespace	10	0
	Transaction Ratio	20	80
	Lock Ratio	20	75
Space Health Index	Tablespace Free Chunk Count	20	0
	Tablespace Free Space Percentage	80	0
Contention Health Index	Free Lists Waits	10	0
	Parallel Query Busy Ratio	10	75
	Rollback Contention Ratio	10	1
	Buffer Busy Wait Ratio	20	1
	Users Blocked	30	0
	Latch Miss Ratio	10	0
Memory Health Index	Redo Log Space Wait Time	10	5
	Percentage of Shared Pool Free	10	15
	Memory Sort Ratio	20	95
	Dictionary Cache Hit Ratio	20	90
	Library Cache Hit Ratio	20	95
	Buffer Cache Hit Ratio	30	90

Microsoft SQL Server Health Index Template

Category	Statistic	Weight	Optimum Value
Databases Health Index	Offline Databases	20	0
	Suspect Databases	30	0
	Errors in Current Log	20	0
	Databases Needing Backup	30	0
Space Health Index	Transaction Logs Low on Space	60	0
	Databases Low on Space	40	0
Memory Health Index	Log Cache Hit Ratio	25	90
	Procedure Plan Hit Ratio	25	90

Category	Statistic	Weight	Optimum Value
	Adhoc SQL Hit Ratio	15	80
	Buffer Cache Hit Ratio	35	90
Contention Health Index	I/O Busy	20	75
	CPU Busy	25	75
	Latch Waits	10	0
	Deadlocks	15	0
	Users Blocked	30	0
Users Health Index	Failed Auto-Param Attempts	30	0
	SQL Recompilations	70	10

Sybase Health Index Template

Category	Statistic	Weight	Optimum Value
Databases Health Index	Databases With Suspect Pages	30	0
	Suspect Databases	40	0
	Users Blocked	30	0
Space Health Index	Transaction Logs Low on Space	50	10
	Databases Low on Space	50	10
I/O Health Index	I/O Busy Rate	30	95
	I/O Error Rate	30	1
	Delayed I/Os	30	10
	Outstanding I/Os	10	10
Memory Health Index	Large I/O Denied	20	0
	Clean Buffer Grab Rate	20	90
	Procedure Cache Hit Rate	30	95
	Data Cache Hit Rate	30	95
Contention Health Index	Device Waits	10	5
	I/O Operating System Limit Delays	5	10
	I/O Engine Configuration Limit Delays	5	10
	I/O Server Configuration Limit Delays	5	10
	Disk I/O Structure Delays	5	10
	Blocking Lock Rate	40	5
	Deadlock Percentage	30	1
Network Health Index	Packet Errors	40	5
	Network Delays	60	5

DB2 Health Index Template

Category	Statistic	Weight	Optimum Value
Instance Health Index	Monitor Heap Utilization	20	80
	Agent Utilization	40	80
	Sort Heap Utilization	40	80
Space Health Index	Tablespaces Low on Space	40	0
	Active Log Utilization	40	0
	Inaccessible Containers	10	0
	Abnormal State Tablespaces	10	0
Memory Health Index	Sort Heap Utilization	20	80
	Lock List Utilization	20	80
	Hash Join Overflow Percentage	5	5

Category	Statistic	Weight	Optimum Value
	Sort Overflow Percentage	5	5
	DB Heap Utilization	20	80
	Package Cache Hit Ratio	10	95
	Catalog Cache Hit Ratio	10	95
	Bufferpool Hit Ratio	10	95
Contention Health Index	Applications Waiting on Locks	50	3
	Lock Timeouts	25	3
	Deadlocks	25	3
Users Health Index	Connection Entry Utilization	80	80
	Agents Waiting for Token	20	3

Recorded Statistic Template Maintenance

The Recorded Statistic Template Maintenance dialog displays the properties of tracked statistics. You can use this dialog to [manipulate templates](#) for the available statistics. On the Maintenance menu, you can open the Recorded Template Statistic Maintenance dialog where you can track all statistics for all of your datasources.

For information on maintaining recorded statistics, see [Recorded Statistic Templates](#).

Configuring Performance Center Client

In the Client Options Editor you can customize the application configuration for your specific needs. The table below describes the Performance Center Client Options Editor tabs:

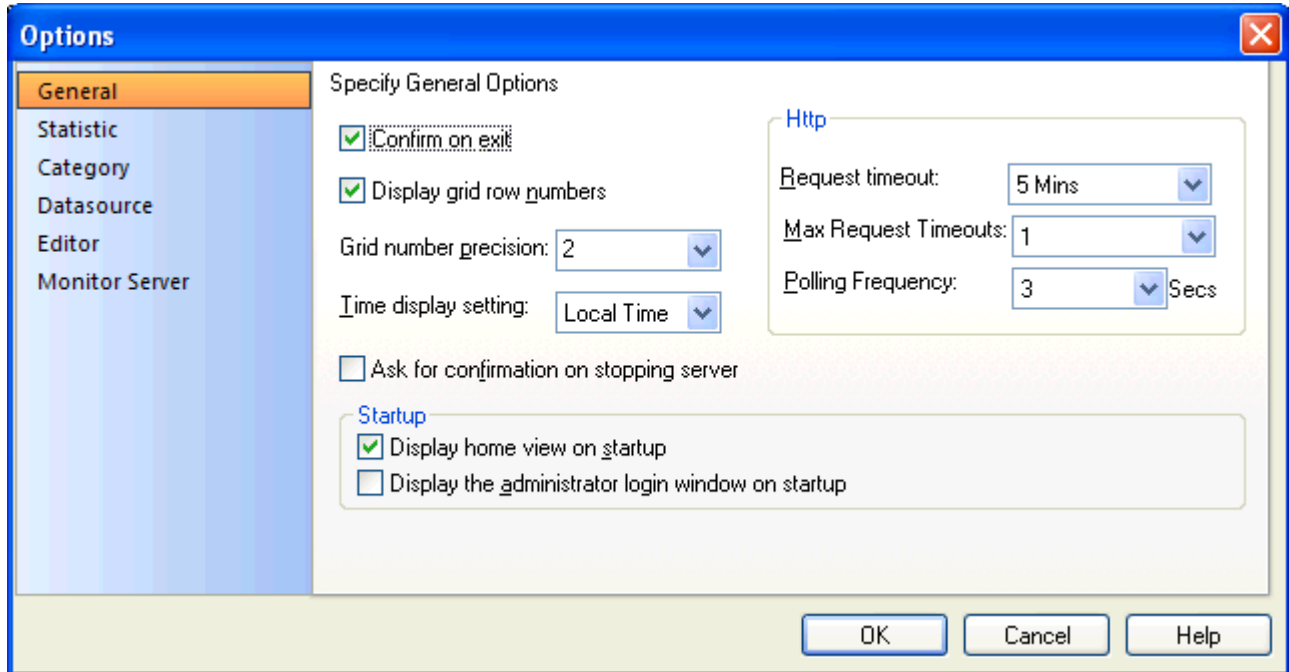
Tab	Description
General	Configure general application options.
Statistic	Specify statistic options.
Category	Specify category options.
Datasource	Specify datasource options.
Editor	Specify SQL Editor options.
Monitor Server	Specify monitor options.
Licensing	Specify licensing options.

For information on configuring the Performance Center Server, see [Configuring Performance Center Server](#).

Opening the Performance Center Client Options Editor

- Click **Monitor > Options**.

Performance Center opens the Options Editor.



General Options

In the General tab of the Options Editor you can configure general application options. Before you can set General options, you must [open](#) the Options Editor. The table below describes General options:

Option	Description	Default
Specify General Options		
Confirm on exit	Displays a confirmation message box when closing Performance Center.	selected
Display grid row numbers	Displays row numbers in all Performance Center view grids.	selected
Grid precision number	Specifies the precision of the row numbers in all Performance Center view grids.	selected
Time display setting	Sets the time that displays on the views.	local time
Ask for confirmation on stopping server		
Startup		
Display home view on startup	Displays the Home view when you open Performance Center.	selected
Display the administrator login window on startup	Displays the Administrator Login dialog when you open Performance Center.	selected
Http		

Option	Description	Default
Request timeout	Specifies the amount of time to wait for an http request before a timeout.	5 minutes
Max Request Timeouts	Specifies how many times Performance Center Client can time out before going into shutdown mode. When Performance Center Client goes into shutdown mode it disables all datasources closes all views. NOTE This value should always be set to 1 unless the Request Timeout is very small (30 seconds or less).	1
Polling Frequency	Specifies the frequency at which the Performance Center Client requests data from the Performance Center Server. NOTE If Performance Center Server monitors many datasources and the Polling Frequency value is 3 seconds, the Performance Center Client may lose messages because the Performance Center Server is producing too much data. If this happens, you can decrease the Polling Frequency to 2 or 1 second(s). However, decreasing the Polling Frequency causes the Performance Center Client to consume more CPU.	3 seconds

Setting General Options

- 1 Click **Monitor > Client Options** and Performance Center opens the Options Editor.
- 2 If necessary, click the **General** tab.
- 3 Select the appropriate options and click **OK**. Performance Center accepts your selections and closes the Options Editor.

Statistic Options

In the Statistic tab of the Options Editor you can specify statistic options. Before you can set Statistic options, you must [open](#) the Options Editor. The table below describes Statistic options:

Option	Description	Default
Foreground	Changes the refresh rate for statistics on Performance Center views.	5 minutes
Drill Down	Changes the refresh rate for statistics on drill down views. NOTE Changing the refresh rate only affects views that you open after making the change. Therefore, views that are already open are not affected.	15 minutes
Show thresholds as pulses	Pulsates the values for all statistics that exceed their thresholds.	selected

Setting Statistics Options

- 1 Click **Monitor > Client Options**. Performance Center opens the Options Editor.
- 2 If necessary, click the **Statistic** tab.
- 3 Select the appropriate options and click **OK**. Performance Center accepts your selections and closes the Options Editor.

Category Options

In the Category tab of the Options Editor you can specify category options. Before you can set Category options, you must [open](#) the Options Editor. The table below describes the Category option:

Option	Description	Default
Show thresholds as pulses	Pulsates the values for all categories that exceed their thresholds.	selected

Setting Category Options

- 1 Click **Monitor > Client Options** and Performance Center opens the Options Editor.
- 2 If necessary, click the **Category** tab.
- 3 Select or clear the option and click **OK**. Performance Center accepts your selections and closes the Options Editor.

Datasource Options

The Datasource tab of the Options Editor allows you to specify datasource options. Before you can set Datasource options, you must [open](#) the Options Editor. The table below describes Datasource options:

Option	Description	Default
Close views associated with disabled datasource	Closes all Performance Center views when the datasource is disabled. Clear this option to view the state of the datasource prior to disabling.	selected
Close views associated with unregistered datasource	Closes all Performance Center views when the datasource is unregistered. Clear this option to view the state of the datasource prior to unregistering.	selected

Setting Datasource Options

- 1 Click **Monitor > Client Options** to open the Options Editor.
- 2 If necessary, click the **Datasource** tab.
- 3 Select or clear the appropriate options and click **OK**. Performance Center accepts your selections and closes the Options Editor.

Editor Options

In the Editor tab of the Options Editor you can specify SQL Editor options. Before you can set SQL Editor options, you must [open](#) the Options Editor. The table below describes the SQL Editor options:

Option	Description	Default
Formatting		
Auto Indent	Formats using an auto-indent.	selected
Expand Tabs	Expands Tabs as part of the formatting.	selected
Tab Size	Select the tab size	4
Appearance		
Syntax Highlighting	Highlight syntax. To select the target font, click the Editor Font list.	selected

Option	Description	Default
Line Numbers	Shows line numbers.	selected

Setting SQL Editor Options

- 1 Click **Monitor > Client Options** and the Options Editor appears.
- 2 If necessary, click the **Editor** tab.
- 3 Select or clear the appropriate options and click **OK**. Your selections are accepted and the Options Editor closes.

Monitor Server Options

In the Monitor Server tab of the Options Editor you can specify monitor options. Before you can set Monitor Server options, you must [open](#) the Options Editor. The table below describes Monitor Server options:

Option	Description	Default
Disable all datasources when Performance Center Server is shut down	Disables all datasources when the monitor server is shut down. Clear this option to view the state of the datasources prior to shutdown.	selected

Setting Monitor Server Options

- 1 Click **Monitor > Client Options** and the Options Editor appears.
- 2 If necessary, click the **Monitor Server** tab.
- 3 Select or clear the appropriate options and click **OK**. Your selections are accepted and the Options Editor closes.

Configuring Performance Center Server

The Server Options Editor allows you to customize the application configuration for your specific needs. The table below describes the Performance Center Server Options Editor tabs:

Tab	Description
Run As	Specify integrated security options.
Analyzer	Specify analyzer options.
Capture Agent	Specify capture agent options.
Repository	Specify repository connection properties.
Datasource Manager	Specify datasource manager options.
Emergency Contact	Specify critical error message properties.
Notifier	Specify notifier options.
Requestor	Specify requestor options.
Report	Specify reporting options.

For information on configuring the Performance Center Client, see [Configuring Performance Center Client](#).

Opening the Performance Center Server Options Editor

- On the **Monitor** menu, click **Server Options**.

TIP: You can also open the Performance Center Server Options Editor through the machine where you installed Performance Center Server. To do so, in the Windows Start menu or on the desktop, open the Performance Center Server program folder, and then click Performance Center Server Options Editor.

Performance Center opens the Performance Center Server Options Editor.

Run As Options

In the Run As tab of the Server Options Editor you can specify the Windows user account that will be used to run the Performance Center Service. To use integrated security the Performance Center Service **MUST** be a Domain User account.

Before you can set Run As options, you must [open](#) the Performance Center Server Options Editor. The table below describes Run As options:

Option	Description
Local System Account	Specify the account that will logon using the system account, rather than a user account. Most services log on to a system account.
Domain User Account	Assigns a logon account to the Performance Center Server.
User name	The user name for the target Domain User Account.
Password	The password for the target Domain User Account.
Domain	The domain name for the target Domain User Account.

For detailed information, see [Operating System View](#).

Setting Run As Options

- On the **Monitor** menu, click **Server Options**.

Performance Center opens the Performance Center Server Options Editor.

- If necessary, click the **Run As** tab.

Performance Center opens the Performance Center Server Options Editor to the Run As tab.

- Select the appropriate options.
- To immediately commit any changes, click **Update**.
- Click **OK**.

Performance Center accepts your selections and closes the Performance Center Server Options Editor.

Analyzer Options

In The Analyzer tab of the Server Options Editor you can specify analyzer options. The Analyzer is an internal process that performs threshold validation. Before you can set Analyzer options, you must [open](#) the Performance Center Server Options Editor. The table below describes Analyzer options:

Option	Description	Default
Enable	Enables the Performance Center Analyzer.	selected
Record Events	Writes a log record for each alert and notification.	selected
Auto Run Statistics	Evaluates the list of default statistics.	selected
Server Display Time Setting	Sets the target time setting for the Performance Center Server.	local time
Send notifications even if action result is empty	Sends any user-defined notifications.	selected

Setting Analyzer Options

- 1 On the **Monitor** menu, click **Server Options**.
Performance Center opens the Performance Center Server Options Editor.
- 2 If necessary, click the **Analyzer** tab.
Performance Center opens the Performance Center Server Options Editor to the Analyzer tab.
- 3 Select the appropriate options.
- 4 Click **OK**.
Performance Center accepts your selections and closes the Performance Center Server Options Editor.

Capture Agent Options

In the Capture tab of the Server Options Editor you can specify capture agent options. The Capture Agent is an internal process that retrieves values for a certain list of statistics and then writes them to the repository. Before you can set Capture Agent options, you must [open](#) the Performance Center Server Options Editor. The table below describes Capture Agent options:

Option	Description	Default
Enable Record Statistical Data	Enables the Performance Center Capture Agent.	selected
Fast Insert	Use the DBMS-specific fast insert.	selected
Capture Settings	Specify the capture times: Insert time-out, capture interval, and maximum batch size.	60 seconds 60 seconds 3000 rows
Record Settings	Specify the record times: record interval and record when specific percent has changed. The Record Now button lets you immediately start recording.	6 hours 1% changed
DMBS tabs	Specify the location of the DBMS-specific file.	n/a

Setting Capture Agents Options

- 1 Click **Monitor > Server Options** and the Server Options Editor. appears
- 2 If necessary, click the **Capture Agent** tab. The Server Options Editor to the Capture Agent tab appears.

- 3 Type the correct information and click **OK**. Your selections are accepted and the Server Options Editor closes.

Repository Options

In the Repository tab of the Server Options Editor you can specify connection properties between the Performance Center Repository and Performance Center. Before you can set Repository options, you must [open](#) the Performance Center Server Options Editor. The table below describes Repository options:

Option	Description	Default
Connection	<p>Displays the ODBC connection string that Performance Center Client uses to connect to the Performance Center Repository.</p> <p>The syntax of the string is: DRIVER={<i>driver_name</i>};SERVER=<i>server_name</i>;UID=<i>user_id</i>;PWD=<i>password</i>; DRIVER is the name of the ODBC Driver (for example, Microsoft ODBC for Oracle). You should not change this value unless directed by Embarcadero Technical Support. SERVER is the name of the datasource. UID is a user ID. PWD is the password corresponding to the user ID, or an empty string if there is no password for the user ID.</p>	<i>local machine</i>
Min Connections	Displays the minimum of connections allowed to the Performance Center Server.	1
Max Connections	Displays the maximum of connections allowed to the Performance Center Server.	10
Release Decay	Displays the time period after which the server releases an inactive connection.	120
Auto-Sync Repository Changes	<p>If you select this option, the server will attempt to periodically synchronize any externally-initiated repository changes (i.e., not via the Performance Center Client or Server applications; for example, calling a stored procedure directly that changes our data in some way).</p> <p>By default, this option is cleared and should remain so for most installations as it does incur an overhead cost even when no changes have occurred. It is only necessary to enable this feature if you know that you will change the repository data in an external manner.</p> <p>For information on manually disabling/enabling Oracle datasources, see Manually Disabling/Enabling Oracle Datasources.</p>	not selected

Setting Repository Options

- 1 Click **Monitor > Server Options** and the Server Options Editor appears.
- 2 If necessary, click the **Repository** tab. The Server Options Editor to the Repository tab appears.
- 3 Select the appropriate options.

TIP: For information on manually disabling/enabling Oracle datasources, see [Manually Disabling/Enabling Oracle Datasources](#).
- 4 Click **OK** and your selections are accepted. The Server Options Editor closes.

Datasource Manager Options

In the Datasource Manager tab of the Server Options Editor you can specify datasource manager options. The Datasource Manager is the internal datasource connection pool manager. Before you can set Datasource Manager options, you must [open](#) the Performance Center Server Options Editor. The table below describes Datasource Manager options:

Option	Description	Default
Resource Available Wait Time	Displays the time for a process to wait to acquire a connection.	2
Resource Recovery Wait Time	Displays the recovery wait before releasing a bad connection to the pool. When a connection to the datasource is found to be bad (i.e., the connection was lost due to some error) the connection is blocked from being used by the Performance Center Server internal components for that amount of time. The Resource Recovery Time lets the connection be properly released from the pool.	120
Datasource Shutdown Wait Time	Specifies the time interval that the Performance Center Server continues to connect to a datasource before killing threads. This interval allows datasources doing fetches additional clean up time before killing threads. For killed threads, the session remains on the database until the database cleans up the session.	15
Datasource Collector Bootup Wait Time	Specifies how long the server bootup logic will wait on a datasource before moving on to the next datasource collector. This wait time is necessary so that each datasource is given enough time to read from the Repository when it boots up.	5

Setting Datasource Manager Options

- 1 Click **Monitor > Server Options** and the Server Options Editor appears.
- 2 If necessary, click the **Datasource Manager** tab. The Server Options Editor to the Datasource Manager tab appears.
- 3 Select the appropriate options and click **OK**. Your selections are accepted and Server Options Editor closes.

Emergency Contact Options

In the Emergency Contact tab of the Server Options Editor you can specify critical error message properties. Before you can set Emergency Contact options, you must [open](#) the Performance Center Server Options Editor. The table below describes Emergency Contact options:

Option	Description	Default
Notification Type	The target type of notification, Don't Notify, Email, Text Pager, Net Send, SNMP, Event Log, or SMS.	Email
Address	The target notification address.	varies
Pager ID	The ID of the pager. For text pager notification only.	none
Password	The password for the pager. For text pager notification only.	none
Max Chars/Msg	The maximum number of characters that the alpha paging service can send per message block. For text pager notification only.	80
Baud Rate	The baud rate of the receiving modem of the paging service. For text pager notification only.	300
Delay	The number of seconds to delay before dialing the number that displays on the recipient's pager. For text pager notification only.	5

Option	Description	Default
SMS Domain	The SMS carrier's domain. For SMS notification only. Typically, the e-mail address of the mobile user is the user's phone number followed by the SMS carrier's domain. For example, a message sent to an AT&T phone will have an e-mail address of 18005551212@mobile.att.net. Here are some of the major carriers and their domains: AT&T - mobile.att.net Sprint - messaging.sprintpcs.com Nextel - messaging.nextel.com Verizon - mst.myVZw.com Voicestream - voicestream.net	none
Max Characters per Msg	The maximum number of characters that the SMS provider allows to be transmitted in a single message. For SMS notification only.	160
SNMP Community	The community in which the recipient resides. For SNMP notification only.	public
Suppress Notification(s)	If this is checked, no critical error messages will be sent to the Emergency Contact.	not selected

Setting Emergency Options

- 1 Click **Monitor > Server Options** and the Server Options Editor appears.
- 2 If necessary, click the **Emergency Contact** tab. The Server Options Editor to the Emergency Contact tab appears.
- 3 Select the appropriate options and click **OK**. Your selections are accepted and the Server Options Editor closes.

Notifier Options

In the Notifier tab of the Performance Center Server Options Editor you can specify notifier options. The Notifier is the internal process responsible for sending the [emergency contact](#) messages.

NOTE: You should leave the default settings for the three boxes in the Modem Properties area unless there are problems.

Before you can set Notifier options, you must [open](#) the Performance Center Server Options Editor. The Notifier tab includes three tabs that let you specify the:

- [Email Configuration](#)
- [Modem Configuration](#)
- [Tivoli Configuration](#)

The table below describes Notifier options available on the Email Configuration tab:

Option	Description	Default
Name	Specify the sender's name.	none
E-mail Address	Specifies the e-mail address of the sender.	repository owner's e-mail address

Option	Description	Default
Authentication	Specifies how the password is handled. CRAM-MD5 - This authentication method sends the password to the mail server in encrypted form. Challenge Response Authentication Mechanism is specified in RFC 2195 and uses the MD5 algorithm, specified in RFC 1321 . LOGIN PLAIN - This authentication method sends the password as clear text (i.e. no encryption) to the mail server. AUTH LOGIN - This authentication method encodes the password using the Base64 algorithm before sending it to the mail server.	none
Username	If authentication is required, you can enter the required username.	none
Password	Specifies the password for the account.	none
Bind To	Specify where to bind the notification.	ANY_IP_ADDRESS
Encoding	Choose among encoding types: Western European (ISO); Arabic (Windows); Baltic (Windows); Central European (ISO); Central European (Windows) and other options on the list.	Western European (ISO)
Send all messages MIME encoded	Send all messages in MIME (Multipurpose Internet Mail Extensions) format.	not selected
Send all messages as HTML	Send all messages in HTML format.	not selected
Auto Connect to the Internet	Automatically establishes a connection to the Internet before attempting to send the mail message.	not selected
Host Name	Identifies the name of the host server.	none
Port Number	Identifies the Port number for the host server.	none

The table below describes Notifier options available on the Modem Configuration tab:

Option	Description	Default
Dial String	Specifies the special dial string needed by your modem. Normally, this value is empty. If the value is empty, the Performance Center Notifier will send your modem an ATDT command when it is ready to dial. If only pulse dial tone is available, you must pass "ATDP" in this parameter.	none
Initialization String	Specifies the special dial string needed by your modem. The maximum length of the modem initialization string is 80 characters. Normally, this value is empty. If the value is empty, the Performance Center Notifier will send your modem an "AT + a" return. The modem must then respond with an "OK". If the default does not work, consult the modem manual for a string that will work. If you need to control the time Power Page waits for a connection before giving up, you can do this by passing "TO=20" where 20 is the target number of seconds. An example string is: "TO=20,ATDT". Any time you need a <cr> replace it with the tilde character (~) (i.e. AT&F~). If you need to pause, the hat character (^) will give you a one second pause. For example, "AT&F~^AT~".	none

Option	Description	Default
Termination String	Specifies the modem command that terminates the call. Normally, this value is empty. If the value is empty, the Performance Center Notifier uses a "#,H;" as the termination string. The # is normally used when you have finished touch-toning in the digits to send to a pager. The # instructs the modem to wait two seconds before hanging up. The H instructs the modem to hang up when it is done. The semi-colon puts the modem in command mode and sends an "OK" to the computer when it has finished dialing.	none

The table below describes the Notifier options on the Tivoli Configuration tab:

Option	Description	Default
configuration file	The directions for the postmsg to get the event to Tivoli. The user specifies the configuration file location.	None
opco	The operation code. The user specifies the opco location.	None
class	The name of the event. Classes are case-sensitive.	Embarcadero_Event
source	This is used by the Tivoli Event Console to assign the event to a configured source. e.g. SENTRY is the source for monitoring. The source must be upper case.	SENTRY
Postmsg syntax	The command string the postmsg.exe runs. For example: Postmsg -f %configuration_file% -r %severity% -m "%message%" "%dynamic_args%" opco=%opco% sub_source=ETPmon.exe hostname=%hostname% %class% %source%	-f %configuration_file% e.g. /usra/Tivoli/data/aix027.conf -r %severity%: severity of the threshold violation: low=WARNING, moderate=MINOR, high=CRITICAL, critical=FATAL -m %message%: the notification subject line %opco%: opco entry on the page: KFNA/PMUS/ITSCA, etc. %hostname%: datasource with the alert/violation %class%: class entry on page. %source%: source entry on page. %dynamic_args%: this is replaced with the information the client enters for email/page/ticket fields.
Display the datasource name as the host name	Indicates if the name of the datasource duplicates as the name of the server or not.	None.

NOTE: The directory that contains the postmsg.exe must reside on the PATH environment variable. If the postmsg.exe is not on the PATH, then the notification will not work.

Setting Notifier Options

- 1 Click **Monitor > Server Options** and the Server Options Editor appears.
- 2 If necessary, click the **Notifier** tab. The Server Options Editor to the Notifier tab appears.
- 3 Select the appropriate options and click **OK**. Your selections are accepted and the Server Options Editor closes.

Requestor Options

In the Requestor tab of the Server Options Editor, you can specify requestor options. The Requestor is the internal process that is responsible for collecting datasource data. Before you can set Requestor options, you must [open](#) the Performance Center Server Options Editor. The table below describes Requestor options:

Option	Description	Default
Ping	Sets the time between pinging the Performance Center Server.	300 seconds
Primary Data	Sets the time that Performance Center Server should wait before fetching the most frequently needed data.	300 seconds
Secondary Data	Sets the time that Performance Center Server should wait before fetching the less frequently needed data. This data comprises large result sets.	900 seconds
On Demand Data	Sets the time that Performance Center Server should wait before fetching the drill-down statistics and statistics that will be unscheduled when the view is closed.	900 seconds
OS Data	Sets the time that Performance Center Server should wait before fetching operating system data.	300 seconds
Log Ping	Sets parameters for pinging the log as Always, On Success, Never, or On Failure	Always
Evaluate Derived Query	Executes derived queries when selected.	selected
Get Max Packets	Specifies the number of packets retrieved from the web server queue. NOTE You must reboot the Performance Center Server to save this value.	1024
Data Fetch	Specifies the fetch timeout for the monitored data source.	300 seconds
Flush Cache	Empties the cache by deleting all stored information.	-

Setting Requestor Options

- 1 Click **Monitor > Server Options** and the Server Options Editor appears.
- 2 If necessary, click the **Requestor** tab.
- 3 Select the appropriate options and click **OK**. Your selections are accepted and the Server Options Editor closes.

Report Options

In the Report tab of the Server Options Editor, you can specify reporting options. Before you can set Report options, you must [open](#) the Performance Center Server Options Editor. The table below describes Report options:

Option	Description	Default
Default Report Task 'Run As' settings	Specifies the account to use when running a report.	local system account

Option	Description	Default
Default Report Output Directory	<p>Specifies the location to place your Performance Center reports.</p> <p>When you run reports on the Web Client for data export jobs, the reports are saved here so you can see them later.</p>	Embarcadero\PerformanceCenter\Report Output
HTML Header Template File	Change the report header for reports created on the Web Client or in an html format.	ReportHeader.htm

Setting Report Options

- 1 Click **Monitor > Server Options**.
- 2 If necessary, click the **Report** tab.
- 3 Select the appropriate options and click **OK**. Your selections are accepted and the Server Options Editor closes.

Using Performance Center

Using Performance Center outlines how to use each component of the application. It details each process and procedure by breaking each down into simple step-by-step instructions. For information on the concepts of Performance Center, see the [Performance Center Overview](#). Here you learn to use the following functions:

Section	Description
Datasource Management	Describes the function of datasources, the process of establishing datasource connections, and managing and organizing the datasources in your enterprise.
Statistics	Describes using the Statistics Maintenance dialog to maintain your statistics.
Health Indexes	Describes using the Health Index Properties and Health Index Templates Maintenance dialogs to maintain your Health Indexes.
Addresses	Describes using the Address Maintenance dialog to maintain your addresses for notifications and actions.
Notifications	Describes using the Notification Maintenance dialog to maintain your notifications.
Actions	Describes using the Action Maintenance dialog to maintain your actions.
Schedules	Describes using the Schedule Maintenance dialog to maintain scheduled jobs.
Datasource Alerts	Describes using the Datasource Alert Maintenance dialogs to maintain your datasource thresholds.
Alert Templates	Describes using the Alert Templates Maintenance dialogs to maintain your Alert Templates.
Recorded Statistic Templates	Describes using the Recorded Statistic Templates Maintenance dialogs to maintain statistic recording.
Reports	Describes using the Report Maintenance dialog to generate and schedule historical reports for all monitored datasources.
Satellite Reports	Describes how to use Satellite Reports to export data between datasources and across platforms.
Archive	Describes using the Archive view.
Alarm	Describes using the Alarm view to view alarms for an Performance Center Server.
Alert Log	Describes using the Alert Log to view alert logs for an Performance Center Server.
Health Index	Describes using the Health Index view.
Instance Parameter	Describes using the Instance Parameter view.
Lock	Describes using the Locks view.
Top Sessions	Describes using the Top Session view.
Top SQL	Describes using the Top SQL view.
Trends	Describes using the Trends view to view statistic trends for an Performance Center Server.
SQL Scripting	Describes using the ISQL Window to execute, write, debug, and tune SQL code.

Datasource Management

A datasource is a reusable connection to a specific database on a specific database server. If you are logged into the datasource as an [Administrator](#), Performance Center offers several ways to let you efficiently manage your datasources. The Explorer window displays all of the datasources registered on the target Performance Center Server. This section includes the following topics:

- [Selecting Datasources](#)
- [Managing Datasources](#)
- [Working With Datasources](#)
- [Organizing Datasources](#)

Selecting Datasources

With Performance Center you can select a monitored datasource.

- 1 Select **Datasource** > **Select** and the Select Datasource dialog appears.
- 2 Click the **Datasource** list, and then click the target datasource.

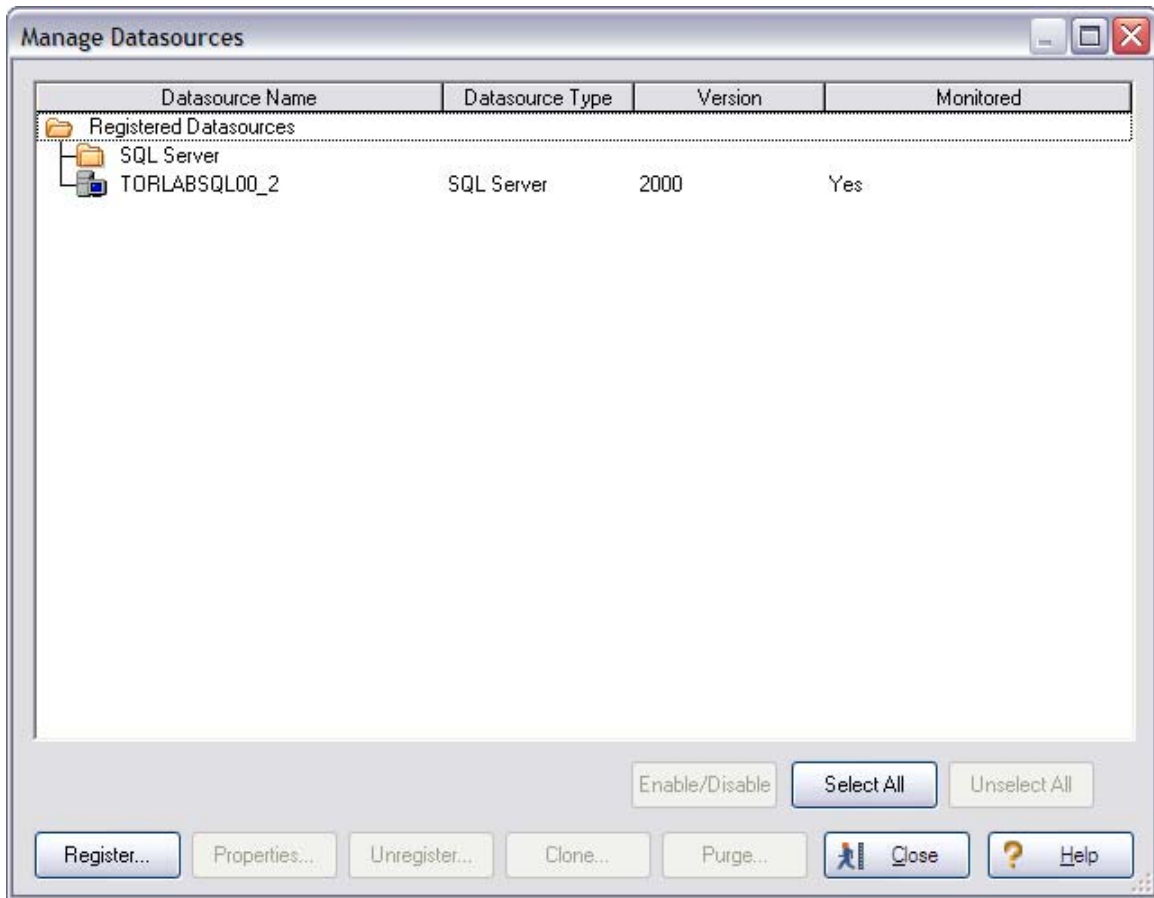
NOTE: For Oracle datasources, Performance Center generates the list of available datasources from the target Performance Center Server's Oracle Tnsnames.ora file.

- 3 Click **OK**.

Managing Datasources

Performance Center offers the Manage Datasources dialog to help you manage your datasources from one location. In the Manage Datasources dialog you can:

- [Open the Register Datasource dialog](#)
- [Open the Clone Datasource dialog](#)
- [Open the Unregister Datasource dialog](#)
- [Open the Datasource Properties dialog](#)
- [Open the Purge Datasource dialog](#)
- [Enable/Disable datasources](#)



The table below describes the options and functionality available in the Manage Datasources dialog:

Column	Description
Datasource Name	Displays all registered datasources.
Datasource Type	Displays the database platform for each registered datasource.
Version	Displays the version of the database platform for each registered datasource.
Monitored	Indicates whether the Performance Center Server is currently monitoring the registered datasource.
Register	Opens the Datasource Properties dialog where you can register every datasource that you want to monitor with the Performance Center Server.
Properties	Opens the Datasource Properties dialog where you can view and modify datasource properties.
Unregister	Unregister the highlighted datasource from the Performance Center Server.
Clone	Opens the Datasource Properties dialog where you can clone the highlighted datasource. Cloning a datasource creates a new datasource by copying all of the properties (thresholds, notifications etc.) from one of your currently registered datasources.
Purge	Opens the Purge Datasource dialog where you can systematically and permanently remove old and unneeded data from the Performance Center Server. Purge is different from delete in that it is possible to regain deleted objects but purged objects are gone forever.
Enable/Disable	Disable or enable Performance Center Server monitoring.

For more information, see:

[Using the Manage Datasources dialog](#)

[Datasource Properties dialog](#)

[Purge Datasource dialog](#)

[Disabling/Enabling a Datasource](#)

Using the Manage Datasources dialog

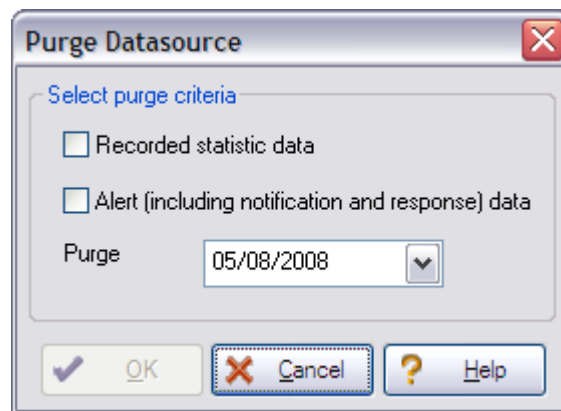
- 1 Select **Datasource > Manage Datasources** and the Manage Datasources dialog appears.
- 2 Select the target datasource and then click the appropriate button.

NOTE: To use the functionality available in the Manage Datasources dialog, you must be logged into the current datasource as an [Administrator](#).

Purge Datasource dialog

The table below describes the options and functionality of the Purge Datasource dialog.

Option	Description
Purge recorded statistic data	Clear all statistics that Performance Center has recorded for the target datasource.
Purge alert (including notification and response) data	Clear the alerts for the target datasource.
If data is <i>x time</i> old	Choose to purge data recorded in a time range: one day, one week, one month, three months, six months, or one year.
If data is recorded on or before	Choose to purge data recorded on or before a specific date.



For more information, see [Completing the Purge Datasource dialog](#).

Completing the Purge Datasource dialog

- 1 In the **Manage Datasources** dialog, select the target datasource(s).
- 2 Click **Purge** and the Purge Datasource dialog appears.

- 3 To clear all the data on statistics that Performance Center has recorded for the target datasource, select **Purge recorded statistic data**.
- 4 To clear all the alerts (including notifications and responses) for the target datasource, select **Purge alert**.
- 5 To specify the time range for the purge, click the appropriate option button. A delete confirmation message box appears.
- 6 Click **Yes** and the target information is purged from the datasource.

Disabling/Enabling a Datasource

The Disable and Enable commands let you disable or enable Performance Center Server monitoring. For example, if you know a database is going to be unavailable because of a forced shutdown for maintenance, you can take advantage of this functionality so you do not receive notification to that effect. There are two ways to disable/enable datasources:

- [Using the Performance Center Client UI](#)
- [Manually, by executing a stored procedure \(ORACLE ONLY\)](#)

NOTE: If you disable a datasource, Performance Center still checks to see whether the datasource and listener are up or down.

Disabling/Enabling Datasources Using the Performance Center Client UI

- 1 In the **Manage Datasources** dialog, select the target datasource, and then click **Disable** or **Enable**.
OR
- 2 On the toolbar, click the datasource list, select the target datasource, click the **Datasource** menu, and then click **Disable** or **Enable**. Performance Center immediately stops or begins monitoring the target datasource.

NOTE: To use the functionality available in the Manage Datasources dialog, you must be logged into the current datasource as an [Administrator](#).

TIP: You can also disable/enable datasources in the [Datasource Properties](#) dialog.

Manually Disabling/Enabling Oracle Datasources

When you execute the SET_DATASOURCE_MONITOR_LEVEL() procedure that follows, you are setting a flag in the repository that tells the Performance Center server the monitoring status of the specified datasource. The server is alerted when the flag value changes so the monitoring status is immediately updated. Once you set the flag to disable monitoring, the Performance Center server no longer tries to poll the target datasource. When you the flag is set to enable, monitoring resumes.

NOTE: There is often a lag in the GUI display, although the change to the repository monitoring status is immediate. Depending on your polling intervals and refresh rates, the client can take several minutes to reflect the changed status. In the event that the client shows a datasource down alert, no corresponding notification is sent.

The following SQL scripts let you manually enable/disable Performance Center monitoring of any registered Oracle datasource.

- 1 On the machine that has the Performance Center Server, on the Windows taskbar, click **Start >Programs, Performance Center > Server Options**. The Performance Center Server opens the Server Options Editor.
- 2 Click the **Repository** and the Server Options Editor opens.
- 3 Select **Auto-Sync Repository Changes** and click **OK**.

4 Execute the appropriate stored procedure against the Performance Center Repository:

- To **DISABLE** a monitored datasource:

```
BEGIN

SET_DATASOURCE_MONITOR_LEVEL( 'DATASOURCENAME' , 0 ) ;

END;
```

- To **ENABLE** a monitored datasource:

```
BEGIN

SET_DATASOURCE_MONITOR_LEVEL( 'DATASOURCENAME' , 7 ) ;

END;
```

Performance Center Server reads the changes to the Repository, and then changes the datasource monitor state accordingly. Depending on the activity of the datasource, this may take up to three minutes.

TIP: You can use Embarcadero DBArtisan's Script Execution Facility, Embarcadero Job Scheduler, or another scheduling application to automate executing these scripts.

Registering/Cloning a Datasource and Viewing Datasource Properties

The Data Source dialog includes three tabs where you can view, modify, or configure basic information on your data sources. The title of the dialog depends on the selected command. For example, if you select Clone from the Data Source menu, the Clone Datasource dialog is displayed. In the Data Source dialog, you can view or configure:

- The data source type.
- The data source name.
- The data source version.
- The user ID and password for connecting to the datasource.
- Performance Center Server monitoring (enable or disable).
- Other data source settings.

The table below describes the Data Source menu commands that you can perform in the Data Source dialog:

Command	Description
Register	Register the data sources you want to monitor with the Performance Center Server.
Clone	Register a new datasource that is a clone of one of your currently registered datasources. Cloning a datasource creates a new datasource by copying all of the properties (thresholds, notifications etc.) from one of your currently registered datasources.
Datasource Properties	View and modify properties of existing registered datasources.

NOTE: Performance Center uses the same dialog for registering datasources, cloning datasources, and viewing or modifying datasource properties. The graphics in the following steps show the datasource registration process.

Using the Datasources dialog

The Datasources dialog includes three tabs. The table below describes the functionality available on each tab:

Tab	Description
General	Configure basic datasource information.
Machine	Specify operating system information for the machine hosting the datasource.
Advanced	Configure connection pools and sampling intervals, and specify the time zone.

For more information, see:

[Completing the Register/Clone/Properties Datasource dialog – General Tab](#)

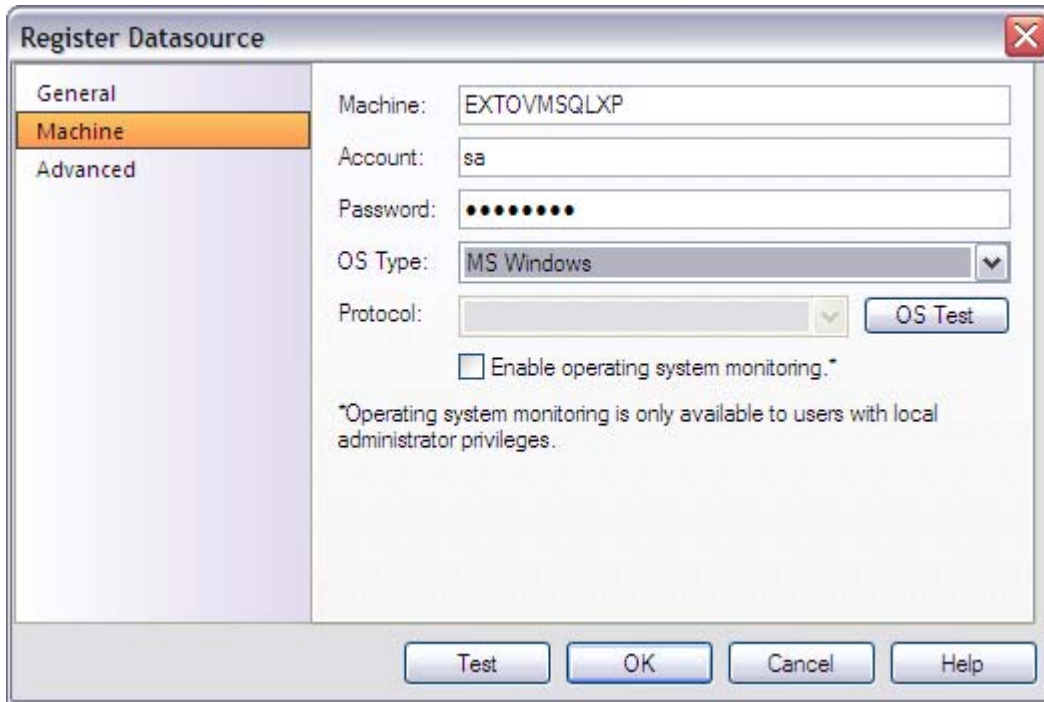
[Completing the Register/Clone/Properties Datasource dialog – Machine Tab](#)

[Completing the Register/Clone/Properties Datasource dialog – Advanced Tab](#)

Register/Clone Datasource dialog – General Tab

The table below describes the options and functionality of the General tab on the Register/Clone/Properties Datasource dialog:

Option	Description
Datasource Type	Select the type of datasource you want to register from the list of available DBMS types.
Datasource Group	Select the datasource group in which you want the datasource to reside.
Datasource Name	Choose the target datasource from the list. When you make your selection, Performance Center opens the Admin. Password for: dialog .
Version Number	Displays the version number of the DBMS software on the datasource.
Login Button	Opens the Admin Password for <datasource>: dialog where you can log into the datasource to retrieve version information.
Enabled for Monitoring	Enable or disable the target datasource for monitoring.
Use OS Authentication	Instructs Performance Center to validate the user's login information with operating system authentication.
Use Database Authentication	Instructs Performance Center to validate the user's login information with database authentication. In the User Name list, enter the target user name, and then in the Password box, type the user's password.
Test Button	Tests the connection to the datasource.



For more information, see:

[Completing the Register/Clone/Properties Datasource dialog – General Tab](#)

[Admin Password for <datasource>: dialog](#)

[Completing the Register/Clone/Properties Datasource dialog – Machine Tab](#)

[Completing the Register/Clone/Properties Datasource dialog – Advanced Tab](#)

Completing the Register/Clone/Properties Datasource dialog – General Tab

- 1 On the **Datasource** menu, click **Register**, **Clone**, or **Properties**. The General tab of the Datasource dialog opens.
- 2 Click the **Datasource Type** list, and then select the target DBMS type. The Datasource Name list is populated with available datasources.
- 3 Click the **Datasource Name** list, and then select the target datasource name. The Admin Password for <datasource>: dialog opens.
- 4 To specify the authentication to use for this connection, in the **Admin Password for <datasource>: dialog**, click one of the options. The table below describes each option:

Option Button	Description
Use OS Authentication	Instructs Performance Center to validate the user's login information with operating system authentication.
Use Database Authentication	Instructs Performance Center to validate the user's login information with database authentication. In the User Name list, enter the target user name, and then in the Password field, type the user's password.

- 5 **OPTIONAL** Click the **Datasource Group** list, and then select the target datasource group.

NOTE: For more information on datasource groups, see [Organizing Datasources](#).

- 6 To configure the Performance Center Server monitor this datasource, select **Enabled for Monitoring**.

- 7 To specify the target user's authentication for all subsequent connections to the new datasource, in the **Connection** box, click one of the option buttons. The table below describes the options:

Option Button	Description
Use OS Authentication	Instructs Performance Center to validate the user's login information with operating system authentication.
Use Database Authentication	Instructs Performance Center to validate the user's login information with database authentication. In the User Name list, enter the target user name, and then in the Password field, type the user's password.

NOTE: The Performance Center server provides the ability to monitor DB2 UDB LUW (Linux, Unix and Windows) instances and databases in your environment. In order to accomplish this it uses the DB2 snapshot facility, numerous API calls, SQL and IBM supplied User Defined Functions. Therefore, you need to register your DB2 monitored data sources with a User ID that has SYSMON, SYSMAINT, SYSCtrl or SYSADM authority.

- 8 To add a new user to the target datasource, click **New** and the target DBMS Login/User Wizard appears.

For information using the Performance Center login/user wizards, see:

[Oracle User Wizard](#)

[SQL Server Login Wizard](#)

[Sybase Login Wizard](#)

- 9 Do one of the following:

Action	Result
To complete the registration or cloning process, click OK .	The datasource is registered or cloned immediately.
To confirm that the datasource connection information is accurate and valid, click Test.	The datasource connection is tested.
To continue configuring the datasource, click the Machine tab .	The Machine tab of the dialog appears. For more information, see Machine Tab .
To continue configuring the datasource, click the Advanced tab .	The Advanced tab of the dialog appears. For more information, see Advanced Tab .

Admin Password for <datasource>: dialog

You must be logged in as an Administrator to register or clone a datasource.

The table below describes the options and functionality of the Admin Password for: dialog:

Option	Description
Use OS Authentication	Instructs Performance Center to validate the user's login information with operating system authentication.

Option	Description
Use Database Authentication	Instructs Performance Center to validate the user's login information with database authentication. In the User Name list, enter the target user name, and then in the Password field, type the user's password.

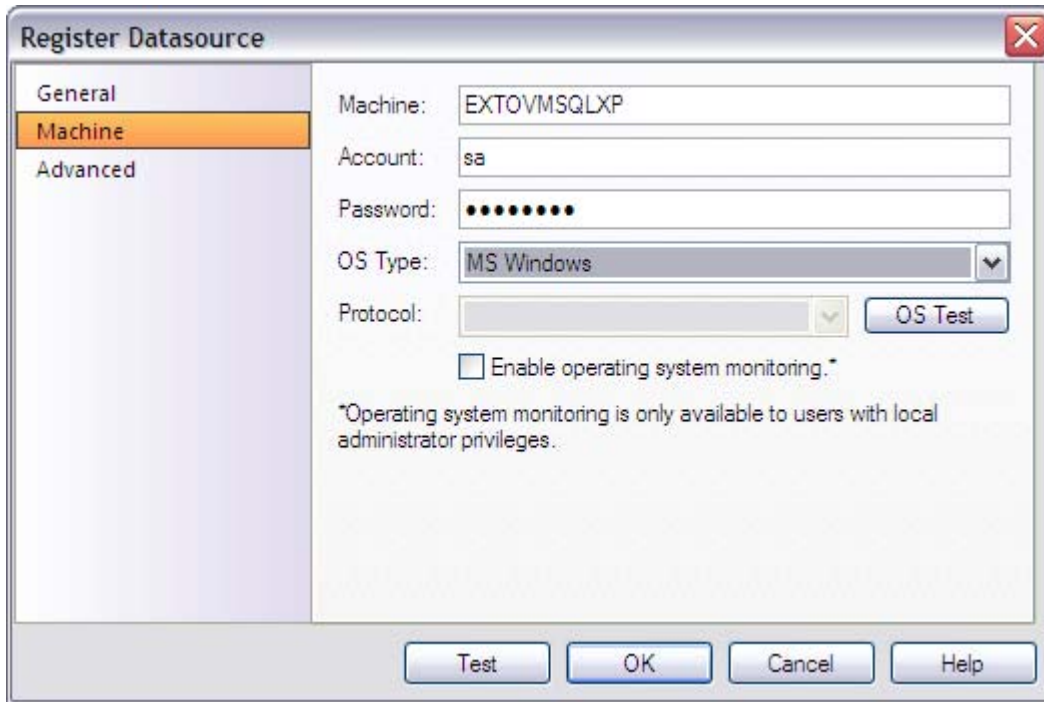


For more information, see [Register/Clone/Properties Datasource dialog – General Tab](#).

Register/Clone/Properties Datasource dialog – Machine Tab

The table below describes the options and functionality of the Machine tab of the Register/Clone/Properties Datasource dialog:

Option	Description
Machine	The name of the machine where the datasource resides.
Account	The name of the account you want to use to log into the machine. You do this in the format DOMAIN/UserName if you are operating in a Windows environment. Unix users log in to the root account. NOTE To use integrated security, the Run As user for the Performance Center Service MUST be a Domain User account. For more information, see Operating System View .
Password	The password for the login account.
OS Type	Select the operating system that is installed on the machine where the datasource resides.
Enable operating system monitoring	Enable or disable Performance Center monitor the operating system so you can see the operating system views.
Protocol	To see operating system views on a Unix or Linux machine, select the appropriate connection protocol: SSH, Telnet, or Rlogin.
OS Test	Retrieve operating system information on demand.



For more information, see:

[Completing the Register/Clone/Properties Datasource dialog – Machine Tab](#)

[Completing the Register/Clone/Properties Datasource dialog – General Tab](#)

[Completing the Register/Clone/Properties Datasource dialog – Advanced Tab](#)

Completing the Register/Clone/Properties Datasource dialog – Machine Tab

- 1 On the **Machine** tab, in the **Machine** field, enter the name of the machine where the datasource resides.
- 2 In the **Account** field, type the account to be used to log into the machine. Windows users log in using the format DOMAIN\UserName for Windows users. Unix or Linux users should log in as root. To use integrated security, the "Run As" user for the Performance Center Service must be a Domain User account. For more information, see [Operating System View](#).
- 3 In the **Password** field, type the password associated with the account that is used to log into the machine where the datasource resides.
- 4 In the **OS Type** field, select the type of operating system that is installed on the machine where the datasource resides.
- 5 To configure the Performance Center Server monitor the operating system, select **Enable operating system monitoring**.
- 6 To retrieve the operating system information on demand, click **OS Test**. The operating system information is retrieved.

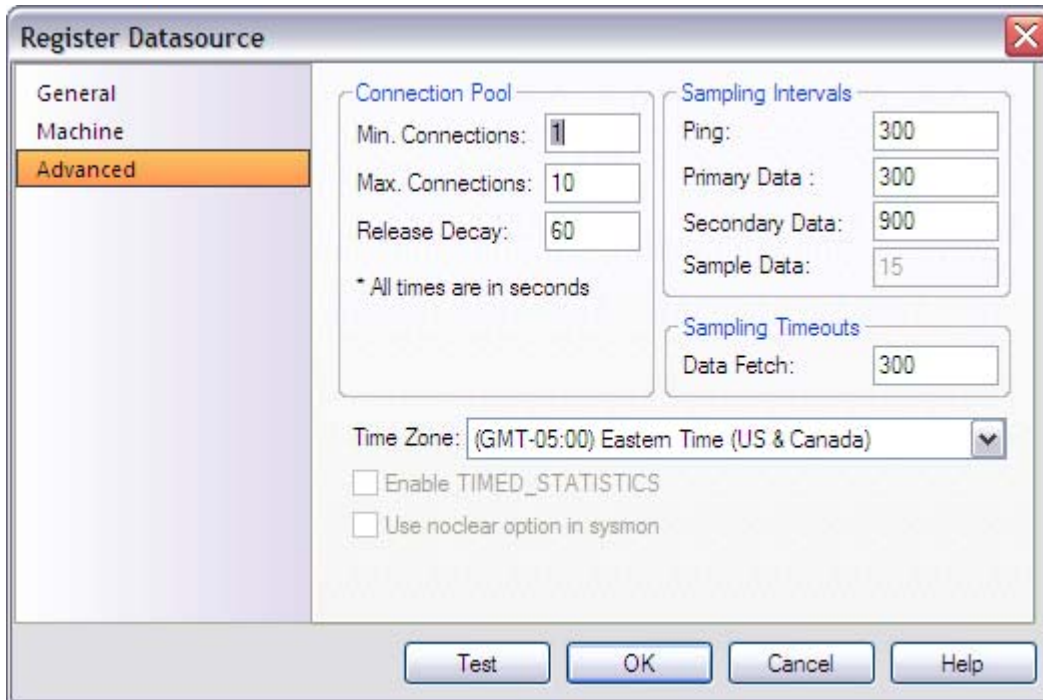
7 Do one of the following:

Action	Result
To complete the registration or cloning process, click OK .	The datasource is registered or cloned immediately
To confirm that the datasource connection information is accurate and valid, click Test .	The datasource connection is tested.
To continue configuring the datasource, click the General tab .	The General tab of the dialog appears. For more information, see General Tab .
To continue configuring the datasource, click the Advanced tab .	The Advanced tab of the dialog appears. For more information, see Advanced Tab .

Register/Clone/Properties Datasource dialog - Advanced Tab

The table below describes the options and functionality of the Advanced tab of the Register/Clone/Properties Datasource dialog:

Option	Description
Min. Connections	The minimum number of connections to the target datasource.
Max. Connections	The maximum number of connections to the target datasource.
Release Decay	The time period after which an inactive connection will be released.
Ping	How often should the datasource be pinged for status.
Primary Data	The amount of wait time before fetching the most frequently needed data.
Secondary Data	The amount of wait time before fetching the less frequently needed data.
SYBASE ONLY	The sampling interval used by the stored procedure that Performance Center uses to gather some statistics.
Time Zone	The time zone for the local server.



For more information, see:

[Completing the Register/Clone/Properties Datasource dialog – Advanced Tab](#)

[Completing the Register/Clone/Properties Datasource dialog – General Tab](#)

[Completing the Register/Clone/Properties Datasource dialog – Machine Tab](#)

Completing the Register/Clone/Properties Datasource dialog – Advanced Tab

- 1 Click **Advanced** and in the **Min. Connections** field, enter the minimum number of connections to this datasource.
- 2 In the **Max. Connections** field, type the maximum number of connections to this datasource.
- 3 In the **Release Decay** field, type the time period after which an inactive connection will be released.
- 4 In the **Ping** field, type the time interval specifying how often Performance Center should ping the datasource for status.
- 5 In the **Primary Data** field, type the time that Performance Center should wait before fetching the most frequently needed data.
- 6 In the **Secondary Data** field, type the time that Performance Center should wait before fetching the less frequently needed data.
- 7 **SYBASE ONLY:** In the **Sample Data** field, type the value of the sampling interval that is used by the Sybase stored procedure to gather some statistics (sp_sysmon).
- 8 Click the **Time Zone** list, and then click the server local time.
- 9 **ORACLE ONLY:** To enable the TIMED_STATISTICS parameter on this datasource, select **Enable TIMED_STATISTICS**.

10 Do one of the following:

Action	Result
To complete the registration or cloning process, click OK.	Performance Center immediately registers or clones the datasource.
To confirm that the datasource connection information is accurate and valid, click Test.	Performance Center tests the datasource connection.
To continue configuring the datasource, click the General tab .	Performance Center opens the General tab of the dialog. For more information, see General Tab .
To continue configuring the datasource, click the Machine tab .	Performance Center opens the Machine tab of the dialog. For more information, see Machine Tab .

Login/User Wizards

Performance Center provides wizards that let you create users and logins for the each supported DBMS:

- [Oracle User Wizard](#)
- [SQL Server Login Wizard](#)
- [Sybase Login Wizard](#)

SQL Server Login Wizard

In the SQL Server Login Wizard, you can create logins and specify security options without knowing the underlying commands.

On the SQL Server Login Wizard you can:

- Name the login.
- Specify the type of login to create.
- Define a login password.
- Select the databases on which to create users.
- Select the user type.
- Specify database roles.

Important Notes

- None

For more information, see:

- [Opening the SQL Server Login Wizard](#)
- [SQL Server Login Wizard - Panel 1](#)

Opening the SQL Server Login Wizard

- 1 Click **Datasource > Register** and the Register Datasource dialog appears.
- 2 On the General tab, from the **Datasource Type** list select **SQL Server**.

- 3 Select a datasource from the **Datasource Name** list, and then click **Login** and you are connected to the target datasource.
- 4 Click the **New** button to the right of the **User Name** box and the first panel of the Login Wizard appears.

SQL Server Login Wizard - Panel 1

The table below describes the options and functions of the first panel of the SQL Server Login Wizard:

Option	Description
What is the name of the login?	Type the login name.
Specify whether this login account will be for a SQL Server or a Windows NT User or Group.	To make Performance Center verify the login with database authentication, click the SQL Server option button, and then complete the password and confirm boxes. To make Performance Center verify the login with operating system authentication, click the NT User or Group option button. Click the Domain list, and then click a domain.

For more information, see [SQL Server Login Wizard - Panel 2](#).

SQL Server Login Wizard - Panel 2

The table below describes the options and functions of the second panel of the SQL Server Login Wizard:

Option	Description
What should be the login's default database?	Select the default database. Avoid specifying the master database to prevent users from creating objects there inadvertently.
What is the login's default language?	Select the default language. Performance Center automatically defaults to the server's default language.

For more information, see [SQL Server Login Wizard - Panel 3](#).

SQL Server Login Wizard - Panel 3

The table below describes the options and functions of the second panel of the SQL Server Login Wizard:

Option	Description
Select database to create users	Select the appropriate target database options. User Name - If you are adding a user name that is different from the Login ID, type the new name in the User Name box. Database Roles - Select the options corresponding to the target database roles.

Sybase Login Wizard

In the Sybase Login Wizard you create a login and specifying security options without requiring you to know any of the underlying commands.

On the Sybase Login Wizard you can:

- Name the login ID.

- Define a login password.
- Specify a default database and language for the login.
- Grant roles to the login.
- Lock the login upon creation.
- Select databases to create users.
- Select user types, user names, and user groups.
- Select aliases.

For more information, see [Sybase Login Wizard - Panel 1](#).

Sybase Login Wizard - Panel 1

The table below describes the options and functions of the first panel of the Sybase Login Wizard:

Option	Description
What is the ID for the Login?	Enter the login name. Can be up to 30 characters long.
What is the password for the new login?	Enter the password for the new login. Must be between six and 30 characters.

For more information, see [Sybase Login Wizard - Panel 2](#).

Sybase Login Wizard - Panel 2

The table below describes the options and functions of the second panel of the Sybase Login Wizard:

Option	Description
OPTIONAL What is the full name of the login box?	Type the full name for the login.
What should be the login's default database?	Database - Select the default database.
What is the login's default language?	Language - Select the default language. If you do not select one, Performance Center uses the server's default language.

NOTE: You should avoid specifying the master database to prevent users from creating objects there inadvertently.

For more information, see [Sybase Login Wizard - Panel 3](#).

Sybase Login Wizard - Panel 3

The table below describes the options and functions of the third panel of the Sybase Login Wizard:

Option	Description
OPTIONAL Select any roles that you wish to grant to the login from the list below	Select the options that correspond to the roles you want to grant.
Do you want to lock the new login after creating it?	You can lock the new login after Performance Center creates it. Locking the login prevents immediate access.

For more information, see [Sybase Login Wizard - Panel 4](#).

Sybase Login Wizard - Panel 4

The table below describes the options and functions of the fourth panel of the Sybase Login Wizard:

Option	Description
Select databases to create users	<p>Select the options corresponding to the databases.</p> <p>User Type - select the user type to add.</p> <p>If you select dbo and you want to transfer aliases and permissions, click the Transfer Aliases and Permissions option.</p> <p>If you select User, in the User box, click the User Name list, and click the target user name, click the Group list, and then click the target user group.</p> <p>If you select Alias, in the Alias box, click the User Name list, and then click the target user name.</p>

NOTE: The options available on this panel depend on the version of Sybase running on the datasource.

Unregistering a Datasource

You can unregister datasources if you are logged into the target datasource as an [Administrator](#). When you unregister a datasource:

- The datasource is no longer available for monitoring.
- All data associated with the datasource is permanently removed from the Performance Center Repository (including all custom alarms, alerts, the health index, thresholds, notifications, blackouts, and any recorded statistical data).

CAUTION: When you unregister a datasource, all historical and configuration data will be lost. Re-registering the datasource does not bring this information back.

Unregistering a Datasource

There are three ways to unregister datasources:

- 1 Click **Main > Datasource**, and select the target datasource, and then on the **Datasource** menu, click **Unregister**.
OR
On the **Enterprise** view, select the target datasources, and then on the **Datasource** menu, click **Unregister**.
OR
In the **Manage Datasources** dialog, click the target datasource(s), and then click **Unregister**.
A confirmation message appears.
- 2 Click **Yes** and the confirmation box closes and immediately unregisters the datasource.

Working with Datasources

Performance Center offers the following functionality when working with datasources:

- [Register](#)

- [Unregister](#)
- [View Properties](#)
- [Clone](#)
- [Purge](#)
- [Enable](#)
- [Disable](#)
- [Organize](#)

You can access this functionality, through several places in the application. The [Manage Datasources dialog](#) offers a central location for accessing this functionality.

NOTE: The organizing features are only available from the Explorer and the Datasource menu.

Organizing Datasources

You can organize the datasources in your enterprise to help you work more efficiently. In the Performance Center Explorer, you can create folders that you can use to group related datasources.

To help you further organize your datasources, you can define sets of datasource groups. These sets are called profiles. You can easily switch between profiles to view different datasource groupings. The following topics are in this section:

- [Creating](#) and [Renaming](#) Groups
- [Changing Groups](#)
- [Removing Groups](#)
- [Enable All in a Group](#)
- [Disable All in a Group](#)
- [Creating](#) and [Renaming](#) Profiles
- [Switching](#) and [Removing](#) Profiles

NOTE: To use the datasource group and datasource profile functionality, you must be logged into the current datasource as an [Administrator](#).

Creating and Renaming a Datasource Group

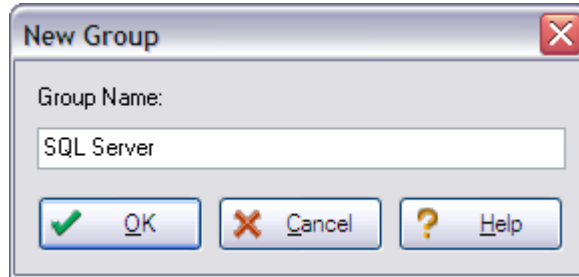
With the New Group and Rename Group commands you can create new datasource groups and rename existing datasource groups. Performance Center uses a similar dialog for these two commands.

- 1 Do one of the following:

On the **Explorer**, select the target parent folder, on the **Datasource** menu point to **Group**, and then click **New Group** or **Rename Group**.

OR

Click **Main Menu > Datasource > Group > New Group**. The New Group or Rename Group dialog appears.

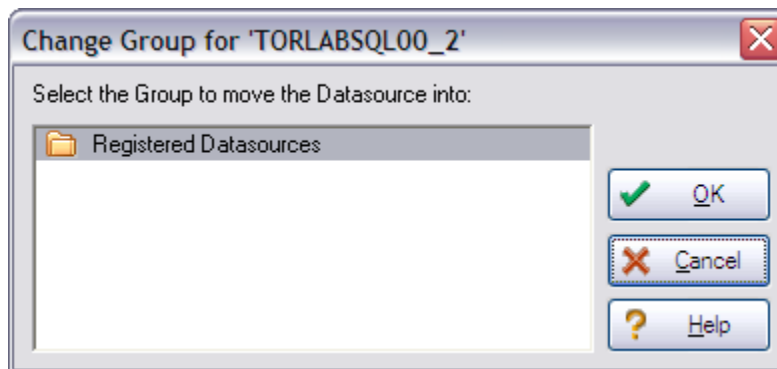


- 2 In **Group Name**, enter the new name of the datasource group.
- 3 Click **OK**. Performance Center adds the new group as a subgroup of the target group in the Performance Center Explorer or updates the Explorer with the renamed datasource group.

Changing a Datasource Group

In the Change Group command you can move a datasource to another datasource group folder.

- 1 Do one of the following:
 - On the **Explorer**, select the target datasource, then click **Datasource > Group > Change Group**.
 - OR
 - Right-click **Explorer and** then click **Change Group**. The Change Group for datasource dialog appears.



TIP: The title of the Change Group dialog includes the name of the target datasource.

- 2 In the box, select the target datasource group and click **OK**. The datasource is moved to the new datasource group in the Performance Center Explorer.

Removing a Datasource Group

In the Remove Group command you can:

- Delete a datasource group and its subgroups, and move all of the datasources in these sub-groups to the root datasource group.
- Delete a datasource group and its subgroups and unregister all of the datasources in the group and any of its sub-groups.

Removing a Datasource Group

Performance Center offers two ways to remove a datasource group:

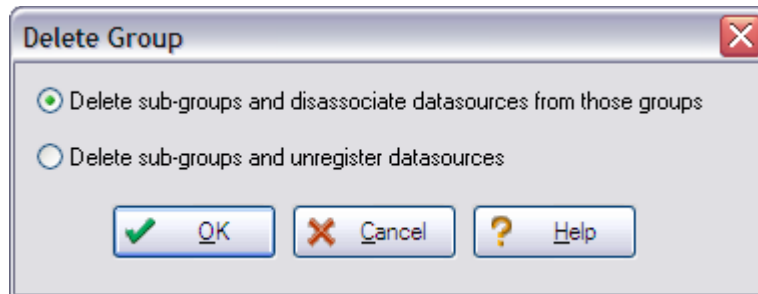
- 1 Do one of the following:

On the **Explorer**, select the target datasource group folder, on the **Datasource** menu point to **Group**, and then click **Remove Group**.

OR

In the **Explorer**, right-click the target datasource group folder, and then click **Remove Group**.

Performance Center opens the Delete Group dialog.



- 2 Click one of the option buttons:

To delete the group, all of its sub-groups, and to disassociate datasources from those groups, click the first option button.

To delete the group, all of its sub-groups, and unregister the datasources, click the second option button

- 3 Click **OK**.

Performance Center immediately deletes the target datasource group.

Enable All the Datasources in a Group

You can quickly enable monitoring for a group of datasources.

- 1 Do one of the following:

On the **Explorer**, select the target datasource group folder, on the **Datasource** menu point to **Group**, and then click **Enable All in Group**.

OR

In the **Explorer**, right-click the target datasource group folder, and then click **Enable All In Group** and a confirmation message box appears.

- 2 Do one of the following:

To enable monitoring for all of the datasources in the target group, click **Yes**.

OR

To cancel the operation and close the box, click **No**.

Disable All the Datasources in a in Group

With the Disable All In Group command, you can quickly disable monitoring for a group of datasources.

- 1 Do one of the following:

On the **Explorer**, select the target datasource group folder, on the **Datasource** menu point to **Group**, and then click **Disable All in Group**.

OR

In the **Explorer**, right-click the target datasource group folder, and then click **Disable All In Group**.

Performance Center opens a confirmation message box.

- 2 Do one of the following:

To disable monitoring for all of the datasources in the target group, click **Yes**.

OR

To cancel the operation and close the box, click **No**.

Creating and Renaming a Datasource Group Profile

The New Profile and Rename Profile commands let you create new datasource group profiles and rename existing datasource group profiles. Performance Center uses a similar dialog for these two commands.

- 1 Do one of the following:

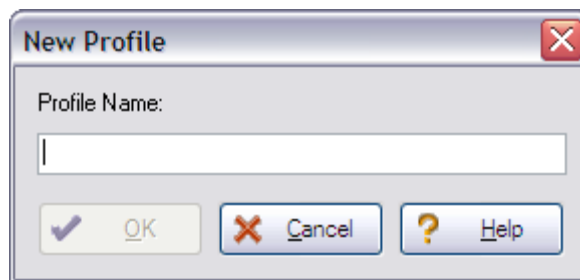
In the **Explorer**, select the highest-level folder. On the **Datasource** menu point to **Profile**, and then click **New Profile** or **Rename Profile**.

OR

On the **Explorer**, right-click the highest-level folder, and then click **New Profile** or **Rename Profile**.

NOTE: The default datasource group profile is Registered Datasources.

The New Profile dialog or the Rename Profile dialog appears.



- 2 In the **Profile Name** box, enter the new name of the datasource group profile and click **OK**. The new datasource group profile is added to the Performance Center Explorer or the Explorer is updated with the renamed datasource profile.

Switching and Removing a Datasource Group Profile

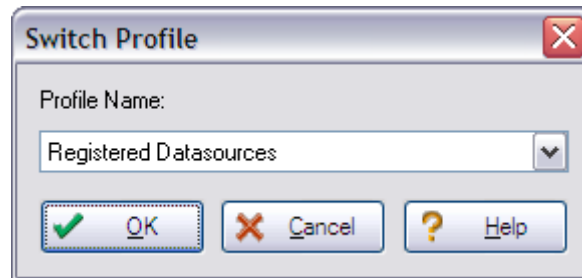
The Switch Profile and Remove Profile commands let you organize the Explorer using another datasource group profile or delete an existing datasource group profile. Performance Center uses a similar dialog for these two commands.

- 1 Do one of the following:

On the **Explorer**, select the highest-level folder, on the **Datasource** menu point to **Profile**, and then click **Switch Profile** or **Remove Profile**.

OR

On the **Explorer**, right-click the highest-level folder, and then click **Switch Profile** or **Remove Profile**. The Switch Profile dialog or the Remove Profile dialog appears.



- 2 In the **Profile Name** box, enter the target datasource group profile and click **OK**. The Explorer switches to the new datasource group profile on the Explorer or deletes the datasource group profile from the Explorer.

Statistics

Statistics are individual performance measurements that Performance Center uses to gauge the overall health of the database. In the Statistics Maintenance dialog you can view all statistics for all monitored datasources.

For more information, see:

[Completing the Statistic Maintenance dialog](#)

[Statistic Properties dialog](#)

[Statistic Maintenance](#)

[Recorded Statistic Templates](#)

Opening the Statistic Maintenance dialog

- 1 Click **Maintenance > Statistics** to open the Statistic Maintenance dialog.

Related Topics

[Completing the Statistic Maintenance dialog](#)

[Statistic Properties dialog](#)

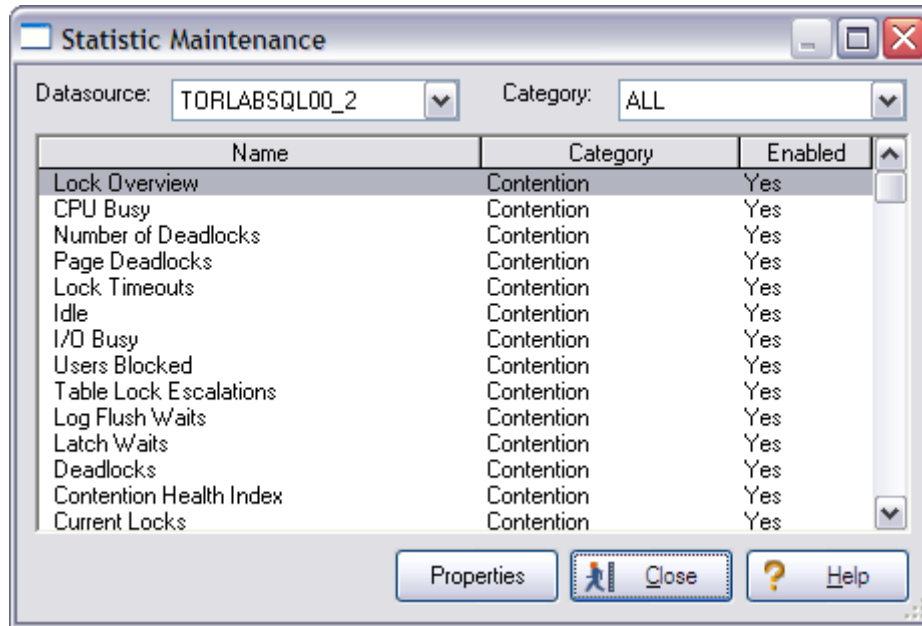
[Statistic Maintenance](#)

[Recorded Statistic Templates](#)

Completing the Statistic Maintenance dialog

The table below describes the options and functionality of the Statistic Maintenance dialog:

Option	Description
Datasource	Select the datasource on which to view statistics.
Category	Select the target category of statistics to view.
Properties	Opens the Statistic Properties dialog where you can view detailed, read-only properties of the target statistic.



For more information, see:

[Opening the Statistic Maintenance dialog](#)

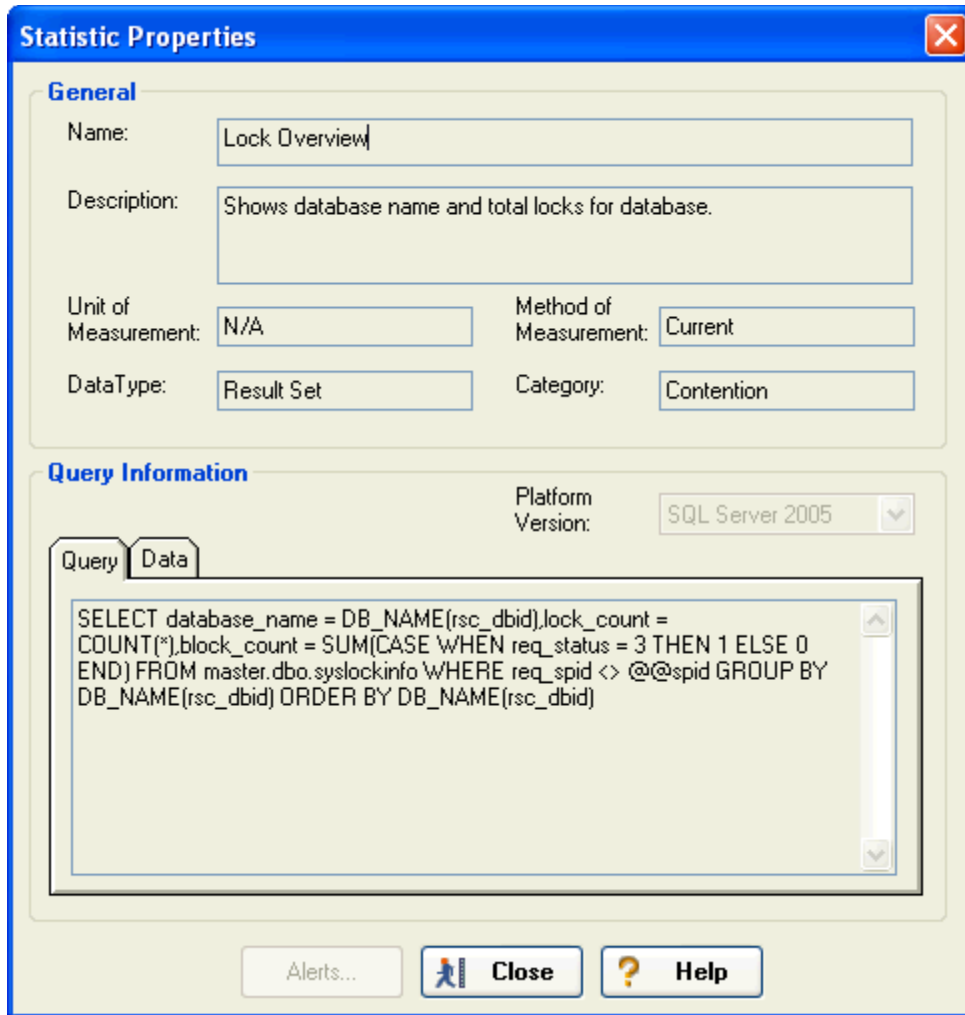
[Statistic Properties dialog](#)

Statistic Properties dialog

The table below describes the options and functionality of the Statistic Properties dialog:

Option	Description
Name	Displays the name of the target statistic.
Description	Provides a brief description of the target statistic.
Unit of Measurement	Displays the unit of measurement for the target statistic.
Method of Measurement	Displays the method of measurement for the target statistic.
DataType	Displays the data type for the target statistic.
Category	Displays the category for the target statistic.
Platform Version	You can see the platform version of the target statistic.

Option	Description
Query tab	You can see the query for the target statistic.
Data tab	Lists the target statistic's value at specific dates and times.
Alerts	Opens the Datasource Alert Maintenance dialog , where you can create, view, and edit thresholds for any statistic.



For more information, see:

[Opening the Statistic Properties dialog](#)

[Statistic Maintenance dialog](#)

[Recorded Statistic Templates](#)

[Datasource Alert Maintenance](#)

Opening the Statistic Properties dialog

- 1 Click **Maintenance > Statistics** and the [Statistic Maintenance dialog](#) appears.
- 2 In the **Statistic Maintenance** dialog, select the target statistic and then click **Properties**.

Health Indexes

Health Index Templates are the logical containers used to define the various performance indexes used to globally measure database performance. To manage your Health Indexes, Performance Center offers the following dialogs:

- [Health Index Properties](#)
- [Health Index Templates](#)

Health Index Properties

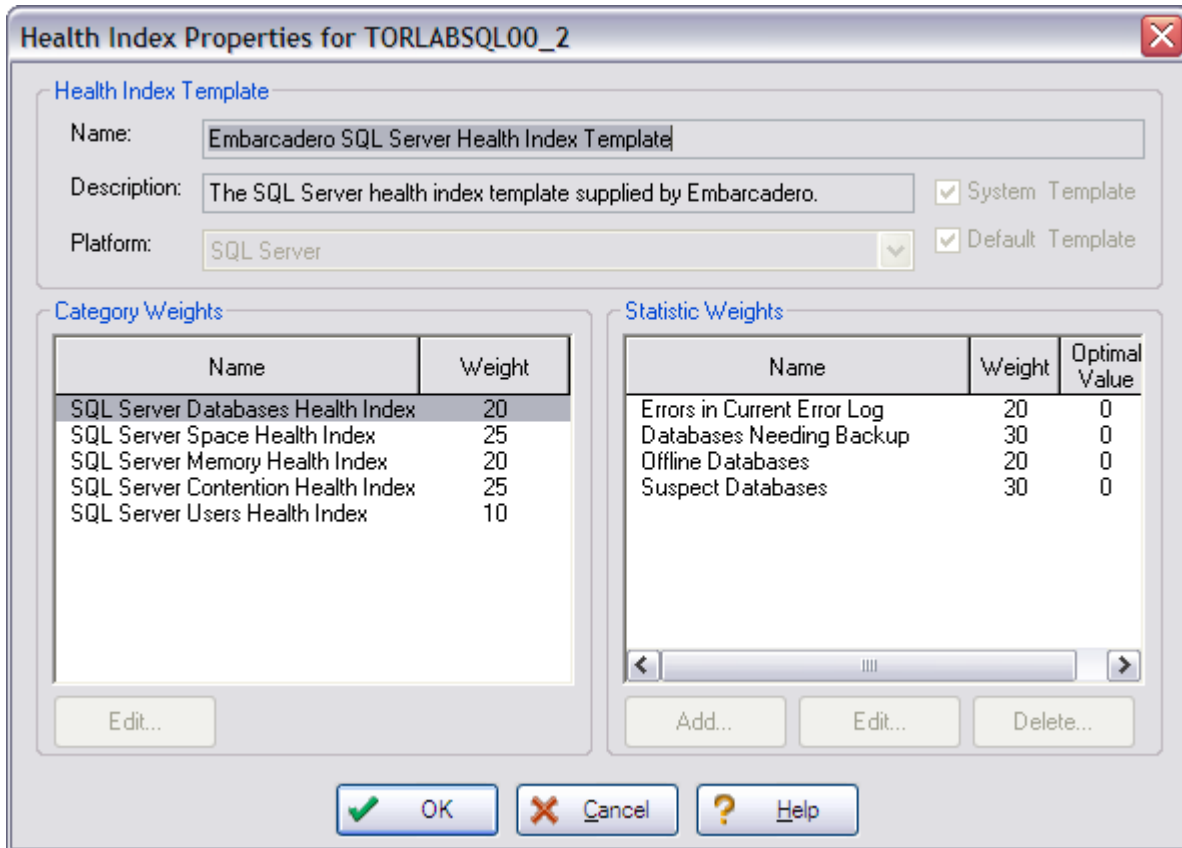
In the Health Index Properties dialog you can view the properties for each [Datasource Health Index Template](#).

Health Index Properties dialog

The table below describes the options and functionality of the Health Index Properties dialog:

Option	Description
Name	Displays the name of the health index template for the target datasource.
Description	Displays the description of the target index template.
Platform	Displays the platform of the target health index template and whether this is the system or default template.
Category Weights	Lets you see the names of the category weights for the target health indexes.
Statistic Weights	Displays the name, weight, and optimal value for the target Category Weight.

TIP: The title of this dialog includes the target datasource.



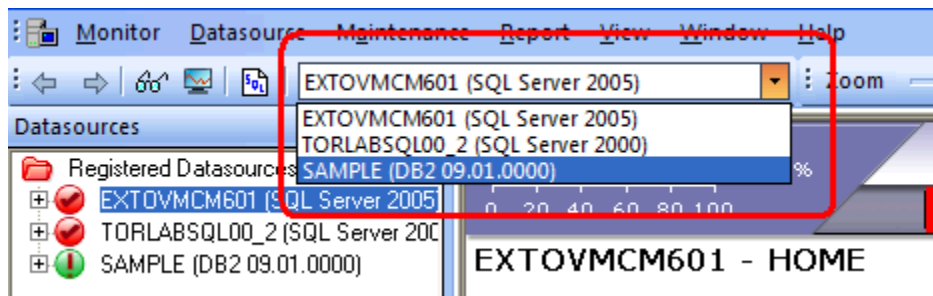
For more information, see:

[Opening the Health Index Properties dialog](#)

[Datasource Health Indexes](#)

Opening the Health Index Properties dialog

- 1 On the **Main** toolbar click the **Datasource** list, and then click the target datasource.



- 2 Click **Datasource > Health Index** to open the Health Index Properties dialog.

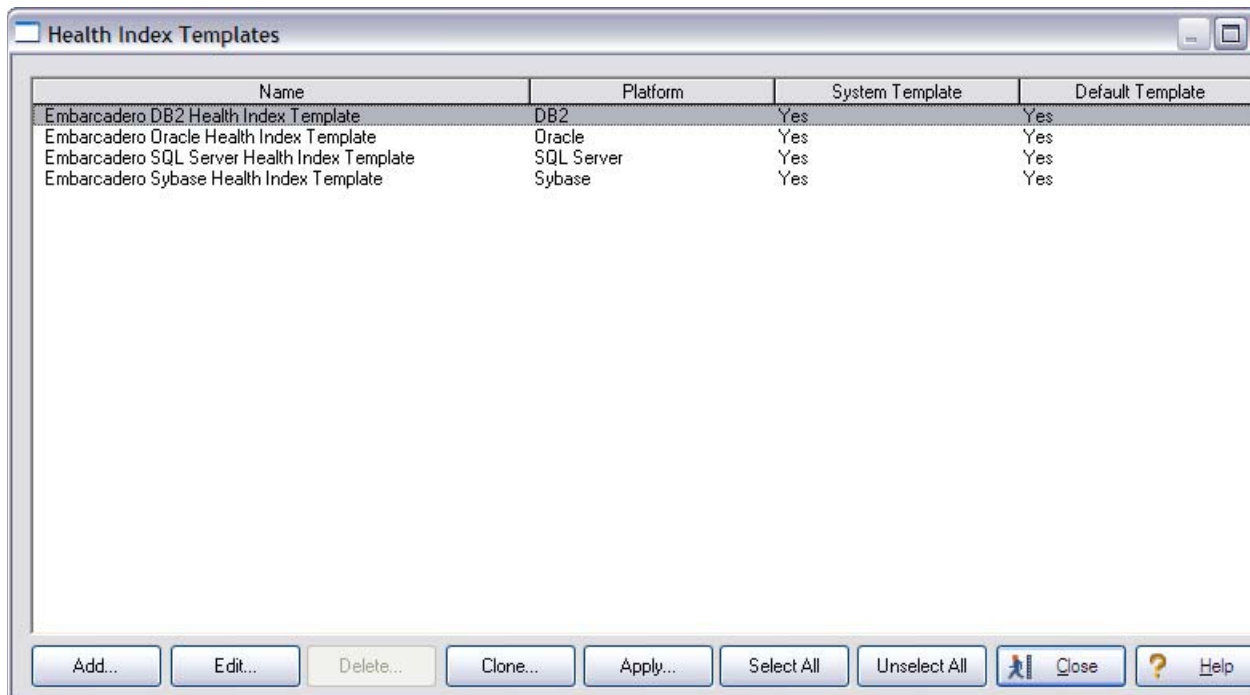
Health Index Templates

The Health Index Templates dialog offers the following functionality for [Health Index Templates](#):

- [Add](#)
- [Edit](#)
- [Delete](#)
- [Clone](#)
- [Apply](#)

The table below describes the Health Index Templates properties you can see in the dialog:

Option	Description
Name	Displays the name of each Health Index Template
Description	Displays the description, if available, of each Health Index Template.
System Template	Yes or No.
Platform	Displays the template's platform.
Default Template	Yes or No.



For more information, see:

[Opening the Health Index Templates dialog](#)

[Adding a Health Index Template](#)

[Editing a Health Index Template](#)

[Deleting a Health Index Template](#)

[Cloning a Health Index Template](#)

[Applying a Health Index Template](#)

[Datasource Health Indexes](#)

Opening the Health Index Templates dialog

- Click **Maintenance > Health Index Templates** to open the Health Index Templates dialog.

For more information, see:

[Adding a Health Index Template](#)

[Editing a Health Index Template](#)

[Deleting a Health Index Template](#)

[Cloning a Health Index Template](#)

[Applying a Health Index Template](#)

[Datasource Health Indexes](#)

Adding/Editing/Cloning a Health Index Template or Viewing Health Index Properties

Performance Center offers the following dialogs where you can view, add, modify, and [clone](#) Health Indexes and Health Index Templates:

- Datasource Health Index Maintenance Add
- Datasource Health Index Maintenance Edit
- Datasource Health Index Maintenance Clone
- Datasource Health Index Properties

NOTE: You must log on to the target Performance Center Server as an [Administrator](#) to add, modify, and clone Health Index Templates.

Related Topic

[Health Index Properties](#)

For more information, see:

[Health Index Template Properties Add dialog](#)

[Health Index Template Properties Edit dialog](#)

[Health Index Template Properties Clone dialog](#)

[Using or Viewing the Health Index Templates or Properties](#)

Health Index Template Properties Add dialog

The table below describes the options and functionality of the Health Index Template Properties Add dialog:

Option	Description
Name:	Enter the name of the health index template you want to add.
Description:	OPTIONAL Lets you type a description of the new health index template.
Platform:	Displays the target DBMS you selected in the Select Platform dialog.
System/Default Template	OPTIONAL To make the new Alert Template a system-defined template, select the System Template option. To make the new Alert Template the default template, select the Default Template option.
Category Weights	Select the category weights you want to add to the new template. Click Edit to make your selections. NOTE The combined total category weight must equal 100.
Statistic Weights	Select target statistics for the template you are creating. Click Add to make your selections.

For more information, see:

[Health Index Template Properties Edit dialog](#)

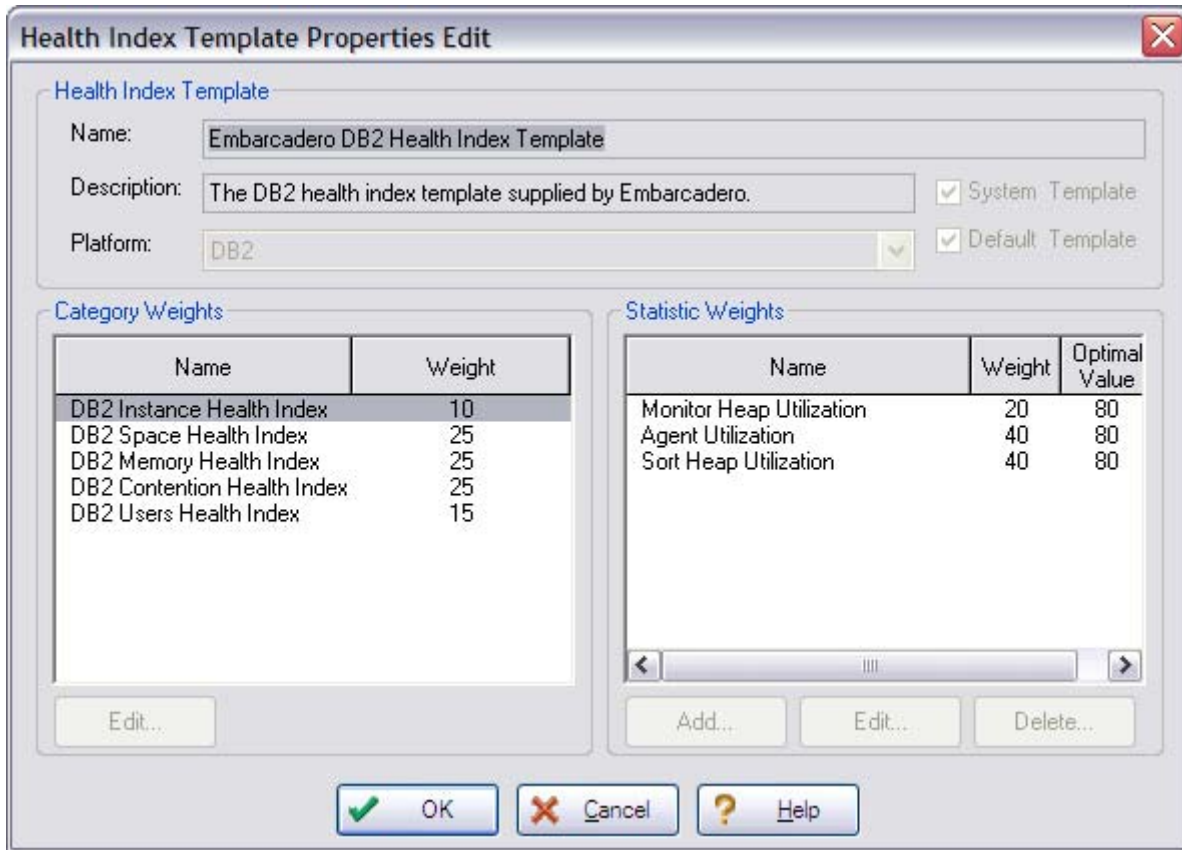
[Health Index Template Properties Clone dialog](#)

[Using or Viewing the Health Index Templates or Properties](#)

Health Index Template Properties Edit dialog

The table below describes the options and functionality of the Health Index Template Properties Edit dialog:

Option	Description
Name:	Make changes to the name of the target health index template.
Description:	OPTIONAL Type or change the description of the target health index template.
Platform:	You can see the DBMS of the target health index template.
System/Default Template	OPTIONAL To make the new Alert Template a system-defined template, select the System Template option. To make the new Alert Template the default template, select the Default Template option.
Category Weights	Change the target template's category weights. To make your selections, click Edit . NOTE The combined total category weight must equal 100.
Statistic Weights	Make changes to statistic weights for the target template. Click Add or Edit to add additional statistic weights or make changes.



For more information, see:

- [Health Index Template Properties Add dialog](#)
- [Health Index Template Properties Clone dialog](#)
- [Using or Viewing the Health Index Templates or Properties](#)

Health Index Template Properties Clone dialog

The table below describes the options and functionality of the Health Index Template Properties Clone dialog:

Option	Description
Name:	Make changes to the name of the target health index template.
Description:	OPTIONAL Enter or change the description of the target health index template.
Platform:	You can see the DBMS of the target health index template.
System/Default Template	OPTIONAL To make the new Alert Template a system-defined template, select the System Template option. To make the new Alert Template the default template, select the Default Template option.
Category Weights	Change the target template's category weights. To make your selections, click Edit . NOTE The combined total category weight must equal 100.
Statistic Weights	Make changes to statistic weights for the target template. To add additional statistic weights or make changes, click Add or Edit .

For more information, see:

[Health Index Template Properties Add dialog](#)

[Health Index Template Properties Edit dialog](#)

[Using or Viewing the Health Index Templates or Properties](#)

Using or Viewing the Health Index Templates or Properties

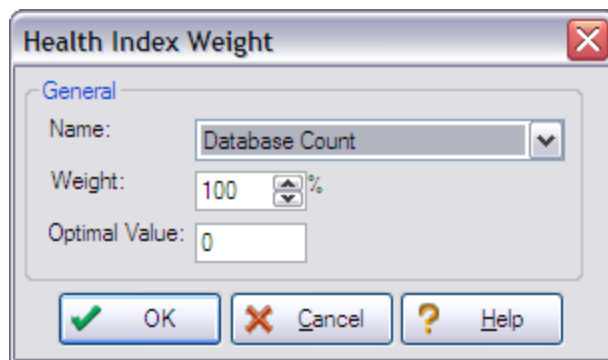
The following steps describe working with the four Datasource dialogs. Use one of the following methods to open the Datasource Health Index Properties dialog:

- 1 Click **Maintenance > Health Index Templates** to open the Health Index Templates dialog.
- 2 In the **Health Index Templates** dialog, click **Edit**.
- 3 In **Name**, enter the name for the Health Index Template.

NOTE: The values in the Health Index Template box can only be modified in the Datasource Health Index Maintenance Add/Edit/Clone dialogs.
- 4 **OPTIONAL** In **Description**, type the description for the Health Index Template.
- 5 Click the **Platform** list, and then click the target DBMS.
- 6 **OPTIONAL** Select one of the following options:

To create the new Alert Template a system-defined template, select **System Template**.

To create the new Alert Template the default template, select **Default Template**.
- 7 In **Category Weights**, select a category, and then click **Edit** to open the Health Index Weight dialog.



- 8 In **Weight**, enter the target weight and click **OK**.
- 9 Repeat steps 7 through 8 for each category in the **Category Weights** box or until you reach a total of 100% for total category weights.
- 10 In **Statistic Weights**, click **Add** and the Health Index Weight dialog appears.
- 11 Click the **Category/Statistic** list, and then click the target statistic.
- 12 In **Optimal Value**, type the value of the statistic in a theoretically perfect situation.
- 13 In **Weight**, enter the target weight.
- 14 Repeat steps 10-13 for each target statistic.

- 15 Click **OK** and you are returned to the Datasource Health Index Maintenance Add/Edit/Clone or Health Index Properties dialog.
- 16 Click **OK** and the dialog closes.

Deleting a Health Index Template

You can delete [Health Index Templates](#) from a Performance Center Server. To delete Health Indexes, you must log on to the target Performance Center Server as an [Administrator](#).

NOTE: You cannot delete the system template.

Deleting a Health Index Template

- 1 Click **Maintenance > Health Index Templates** to open the Health Index Templates dialog.
- 2 In the **Health Index Templates** dialog, select the target Health Index Template(s).
- 3 Click **Delete** and a confirmation message appears.
- 4 In the message box, click **Yes**.

Applying a Health Index Template

Performance Center lets you apply any [Health Index Template](#) to an Performance Center Server. To apply Health Index Templates, you must log on to the target Performance Center Server as an [Administrator](#).

Applying a Health Index Template

- 1 Click **Maintenance > Health Index Templates** to open the Health Index Templates dialog.
- 2 In **Health Index Templates**, select the target Health Index Template(s), click **Apply**, and the Select Datasources dialog appears.
- 3 Select the target datasource(s) and click **OK**. The Health Index Template is applied to the target datasource(s).

Addresses

Addresses are notification recipient information that is stored in the Performance Center [Repository](#). In the Address Maintenance dialog you can view all addresses for all monitored datasources.

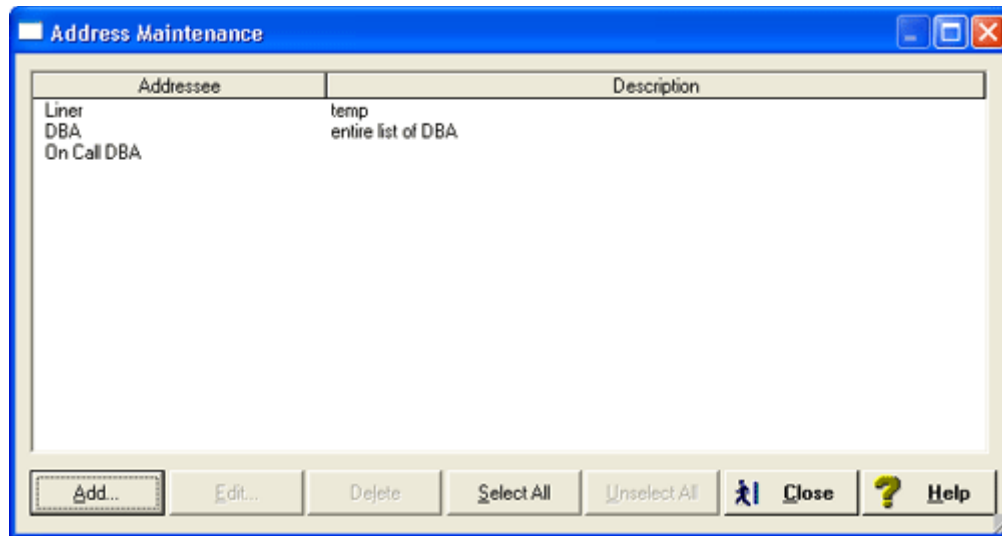
If you are logged on as an [Administrator](#), you have the following functionality for addresses:

- [Add](#)
- [Edit](#)
- [Attach to a Notification](#)
- [Delete](#)

The table below describes the Address Maintenance properties available in the dialog:

Option	Description
Addressee	View the current addressees.

Option	Description
Description	View the description, if any, of each addressee.



For more information, see:

[Opening the Address Maintenance dialog](#)

[Adding an Address](#)

[Editing an Address](#)

[Attaching an Address to a Notification](#)

[Deleting an Address](#)

[Address Maintenance](#)

Opening the Address Maintenance dialog

- Click **Maintenance > Addresses** to view the Address Maintenance dialog.

Adding or Editing an Address

Performance Center lets you add [addresses](#) to the repository, and it lets you edit addresses stored repository. To add or edit addresses, you must log on to the target Performance Center Server as an [Administrator](#).

You add or edit addresses using the Address Properties dialog. The table below describes the options and functionality of the Address Properties dialog:

Option	Description
Name	Enter a new name or change the existing name of the addressee.
Description	OPTIONAL Add or change a description of the addressee.

Option	Description
Type	Displays the type of the target address: alpha page, e-mail, event log, IP, numeric page, SMS, SNMP, or WINS. When you click Add, the Address Detail dialog appears.
Role	Displays the role of the target address: business, home, mobile, pager, or temporary.
Information	Displays the details of the address, such as the complete mail address.

For more information, see:

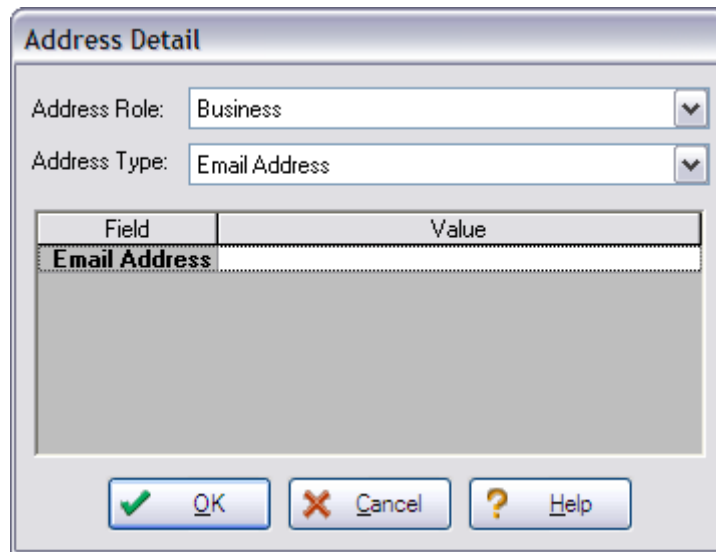
[Address Details dialog](#)

[Completing the Address Properties dialog](#)

Address Details dialog

The table below describes the options and functionality of the Address Details dialog:

Option	Description
Address Role	Select the target role from a list: business, home, mobile, pager, or temporary.
Address Type	Select the target address type, including: alpha page, e-mail address, event log, IP, numeric page, SMS, SNMP, Tivoli eMsg, or WINS. The table below describes the Address Types .
Field	Displays the name of the target Address Type
Value	Enter the target address.



Address Types

The table below describes the address types:

Type	Element	Description	Example
Alpha Page Address	Access Number	The access number that initiates the alphanumeric page.	14158313131
	Baud Rate	The baud rate of the receiving modem of the alpha paging service.	300
	Chars Per Message Block	The maximum number of characters that the alpha paging service can send per message block.	80
	Password	The password required to send an alpha page. This is set by the alpha paging service and is rarely used.	L23456
	Pager ID	The ID of the pager.	1154095
E-mail Address	Mail Address	The e-mail address of the addressee.	user@domain.com
Event Log Address	Event Log Address	<p>The network name of the local or remote computer with the target Windows event log.</p> <p>If the machine where you view the event log does not have Performance Center installed, you will see the following information in the Event Log Description box:</p> <p>The description for Event ID () in Source (ET Notifier) cannot be found. The local computer may not have the necessary registry information or message DLL files to display messages from a remote computer.</p> <p>However, you can still see the Performance Center notification information.</p>	ComputerName
IP Address	IP Address	The IP address of the target machine.	10.10.34.15
	Subnet Mask	The subnet, or address mask.	255.0.4.1
Numeric Page Address	Pager Phone Number	The phone number for the pager.	14158343131
	Pager Delay	The number of seconds to delay before dialing the number that displays on the recipient's pager.	5
SMS Address	Phone Number	Phone number (usually a mobile phone).	14159887371
	Domain	Carrier's SMS domain. This varies by service provider.	messaging.att.com
	Max Message Length	The maximum number of characters allowed in a single message. If the message length exceeds this limitation, the message will be broken up into multiple segments.	240
SNMP Address	SNMP Address	The Simple Network Management Protocol (SNMP) address of the recipient machine.	JimmyG
	SNMP Community	The community in which the recipient resides.	Public NOTE Public is the default. You can configure the community using the Emergency Contact Tab on the Server Options dialog.
Tivoli eMsg	eMsg email_to	List of e-mail recipients.	"user@domain1.com user@domain2.com"

Type	Element	Description	Example
	eMsg email_reqd	Indicates whether or not a message is required as a response to an Embarcadero_Event in YES or NO format. The default is NO. YES/NO must be upper case	NO
	eMsg pager_to	List of pager recipients.	14158343131
	eMsg pager_reqd	Indicates whether or not a page is required in response to an Embarcadero_Event in a YES or NO format. The default is NO. YES/NO must be upper case.	NO
	eMsg ticket_reqd	Indicates whether or not a ticket is required in response to an Embarcadero_Event in a YES or NO format. The default is NO. YES/NO must be upper case	NO
	optional [slot=value]	The variable(s) included as part of an Embarcadero_Event. NOTE String variables, like email_msg, cannot exceed 255 characters.	pager_msg (message to display on pagers); hostname (name of server having the problem)
WINS Address	WINS Name	The Windows Internet Naming Service (WINS) name of the target machine.	production

For more information, see [Completing the Address Properties dialog](#).

Completing the Address Properties dialog

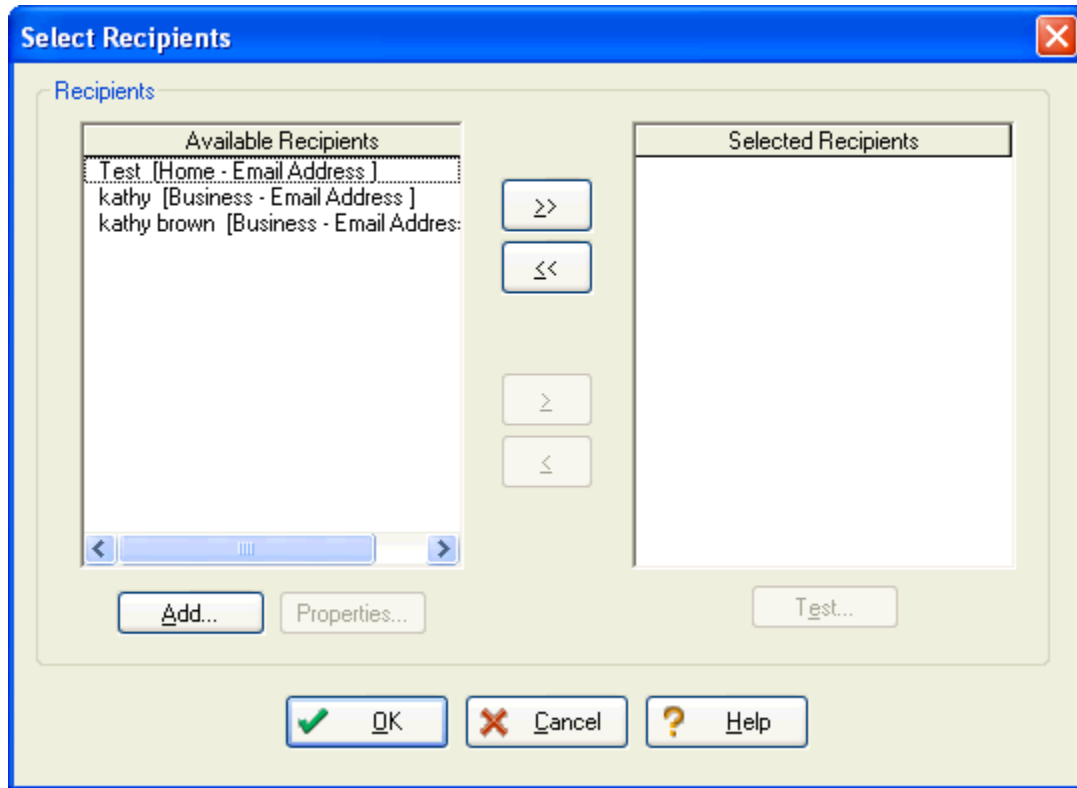
- 1 Click **Maintenance > Addresses** to open the Address Maintenance dialog.
- 2 In the **Address Maintenance** dialog, click **Add** or **Edit** to open the Address Properties dialog.
- 3 In the **Name** box, enter the name for the notification.
- 4 **OPTIONAL** In **Description**, enter the description for this addressee.
- 5 Click **Add** and the Address Detail dialog appears.
- 6 Click the **Address Role** list, and then click the target role.
- 7 Click the **Address Type** list, and then click the target type. Click **OK**. The Address Properties dialog appears.
- 8 To add or edit more addresses, repeat [steps 1-7](#).
- 9 Click **OK**.

Attaching an Address to a Notification

You can attach [addresses](#) to notifications. You must attach addresses to notifications in the Notification Properties dialog. To attach addresses, you must log on the target Performance Center Server as an [Administrator](#).

Attaching an Address to a Notification

- 1 Click **Maintenance > Notifications** and the Notification Properties dialog appears.
- 2 In the **Notification Maintenance** dialog, click **Add** and Select Recipients dialog appears. The Available Recipients box is populated with all of the addresses on the Performance Center Server.



- 3 Do one of the following:

To select the target address(es), click the arrows to move target recipients from **Available Recipients** to **Selected Recipients**.

OR

To add a new address, click **Add** and the Address Properties dialog appears.

OR

To view and/or modify an address, click **Properties** and the Address Properties dialog appears.

NOTE: For information on using the Address Properties dialog, see [Adding or Editing an Address](#).

- 4 When you finish selecting addresses, click **OK**. The address are attached o the notification, the Select Recipients dialog closes, and you are returned to the Notification Properties dialog.
- 5 [Complete](#) the **Notification Properties** dialog.

Deleting an Address

You can delete [addresses](#). To do so, you must log on the target Performance Center Server as an [Administrator](#).

Deleting an Address

- 1 Click **Maintenance > Addresses** to open the Address Maintenance dialog.
- 2 In the **Address Maintenance** dialog, select the target address and click **Delete**. A confirmation message appears.

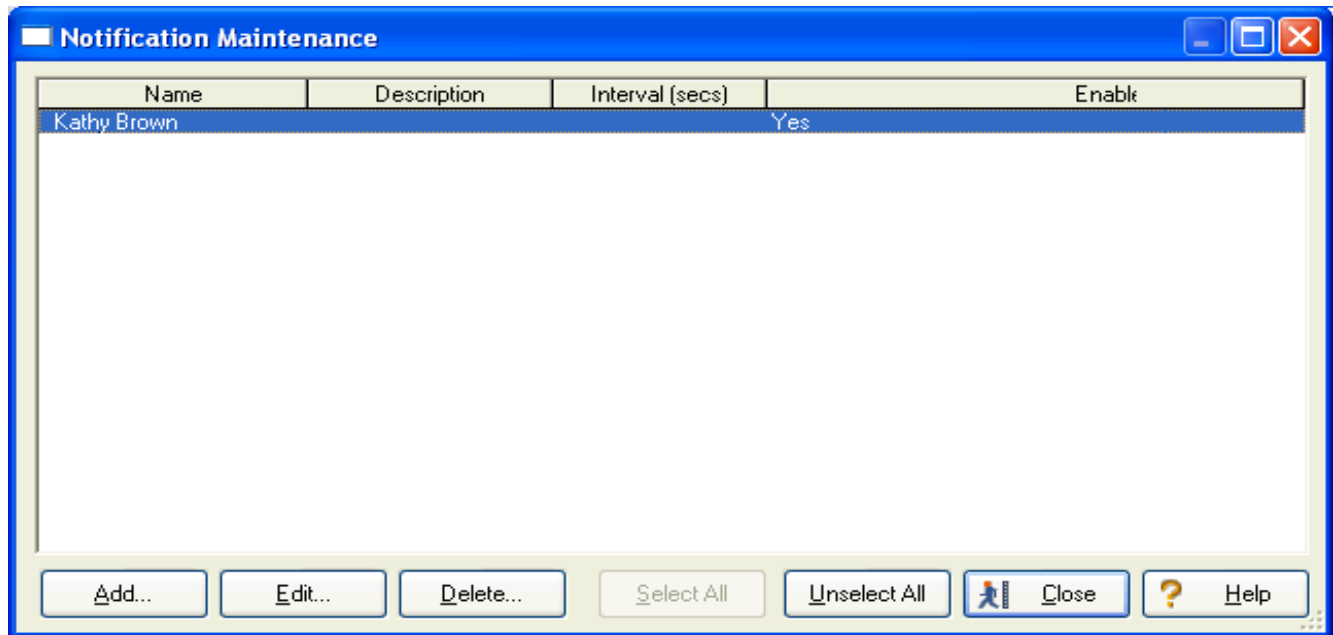
- 3 Click **Yes** to close the confirmation box and delete the address.

Notifications

Notifications are reusable messages that you tell Performance Center to send in response to threshold violations. All notifications are grouped for the current datasource in one dialog. You can view all notifications for all monitored datasources in the Notification Maintenance dialog.

The table below describes the properties you see in the Notification Maintenance dialog:

Option	Description
Notification Name	Displays a list of notification names.
Description	Displays descriptions, if any, attached to each notification address.
Enable	Displays Yes or No, as selected, for each notification address.
Interval (sec)	Displays interval selected for each notification address.



If you are logged on as an [Administrator](#), the following functionality is available for notifications:

- [Add](#)
- [Edit](#)
- [Attach](#)
- [Delete](#)

For more information on notifications, see:

[Adding a Notification](#)

[Tivoli Notifications](#)

[Editing a Notification](#)[Attaching a Notification](#)[Deleting a Notification](#)[Notification Maintenance](#)[Opening the Notification Maintenance dialog](#)

Opening the Notification Maintenance dialog

- Click **Maintenance > Notifications** and the Notification Maintenance dialog appears.

Adding or Editing a Notification

Performance Center lets you add [notifications](#) to the repository, and it lets you edit notifications stored in the repository. To add or edit notifications, you must log on to the target Performance Center Server as an [Administrator](#).

You add or edit notifications using the Notification Properties dialog. The table below describes the options and functionality of the Notification Properties dialog:

Option	Description
Name:	Enter the name for the notification.
Enabled:	Choose to enable the notification.
Description:	OPTIONAL Type or modify the description of the notification.
Interval:	Set the number of minute/seconds to wait before Performance Center sends out the notification.
Subject:	OPTIONAL Type a title for the notification message. To select a predefined variable, click the arrow, and then click an item on the list.
Message:	OPTIONAL Type a message for the notification. To select a predefined variable, click the arrow, and then click an item on the list.
Address	To add an address to the notification, click Add.
Blackout Schedule	To attach a schedule to the notification, click Blackout.

For more information, see:

[Completing the Notification Properties dialog](#)

Tivoli Notifications

[Attaching a Notification](#)[Deleting a Notification](#)[Notification Maintenance](#)

Completing the Notification Properties dialog

- Click **Maintenance > Notifications**. The Notification Maintenance dialog appears.
- In the **Notification Maintenance** dialog, click **Add** or **Edit** and the Notification Properties dialog appears.

- 3 In **Name**, enter the name for the notification.
- 4 To enable the notification on completing these steps, select the **Enabled** option.
- 5 You have the option of entering a notification description In the **Description** box.
- 6 In the **Interval** boxes, type or use the arrows to enter the corresponding minutes and seconds that Performance Center waits before sending notifications.
- 7 You also have the options to enter any information to be sent with the notification in the **Subject** and **Message** boxes.
- 8 **OPTIONAL** To specify a predefined variable, click the arrows next to the **Subject** and **Message** boxes to populate the boxes with the target values.
- 9 Do one of the following:

To create the notification, click **OK**. The Notification is added to the target Performance Center Server and the Notification Properties dialog closes.

OR

To add an address to the notification, click **Add**. The Select Recipients dialog appears. The Available Recipients box is populated with all of the addresses on the Performance Center Server.

NOTE: For information on using the Select Recipients dialog, see [Attaching an Address to a Notification](#).

OR

To attach a schedule to the notification, select an address, and then click **Blackout**. The Select Notification Schedule dialog appears. The Select Notification dialog is populated with all of the schedules on the target Performance Center Server.

OR

- 10 To detach the blackout schedule from the target notification, click **Unblackout**. The blackout schedule from the target notification detaches and the Select Notification Schedule dialog appears.

NOTE: For information on using the Select Notification Schedule dialog, see [Attaching a Schedule to a Notification](#).

Tivoli Notifications

Setting up Tivoli Notifications is a four step process:

- 1 [Verify Tivoli parameters on the Performance Center Server](#)
- 2 [Set up a Tivoli address on the Performance Center Client](#)
- 3 [Create and Test the Tivoli Notification](#)
- 4 [Attach the Tivoli Notification to an Alert](#)

Verifying Tivoli Parameters

- 1 Click **Start > Programs > Embarcadero Performance Center > Server Options** to open the Server Options Editor.
- 2 Click **Notifier** and the Notifier tab of the Server Options Editor appears.
- 3 Click **Tivoli Configuration**.

- 4 Select the appropriate options, and then click **OK**. Your selections are accepted and Server Options Editor closes.

The table below describes the Notifier options on the Tivoli Configuration tab:

Option	Description	Default/Example
configuration file	The directions for the postmsg to get the event to Tivoli. The user specifies the configuration file location.	None
opco	Specify the operation code (opco) location.	None
class	The name of the event. Classes are case sensitive.	Embarcadero_Event
source	This is used by the Tivoli Event Console to assign the event to a configured source, e.g., SENTRY is the source for monitoring. The source must be upper case.	SENTRY
Postmsg syntax	<p>The command string the postmsg.exe runs.</p> <p>NOTE The directory that contains the postmsg.exe must reside on the PATH environment variable. If the postmsg.exe is not on the PATH, then the notification will not work.</p> <p>For example: Postmsg -f%configuration_file% -r %severity% -m "%message%" %dynamic_args% opco=%opco% sub_source=ETPmon.exe hostname=%hostname% %class%%source%</p>	<p>-f %configuration_file%, e.g., /usra/Tivoli/data/aix027.conf</p> <p>- r %severity%: severity of the threshold violation: low=WARNING, moderate=MINOR, high=CRITICAL, critical=FATAL</p> <p>-m %message%: the notification subject line</p> <p>%opco%: opco entry on the page: KFNA/PMUS/ITSCA, etc.</p> <p>%hostname%: datasource with the alert/violation</p> <p>%class%: class entry on page</p> <p>%source%: source entry on page.</p> <p>%dynamic_args%: this is replaced with the information the client enters for email/pager/ticket fields.</p>

Setting Up a Tivoli Address

- 1 Click **Maintenance > Addresses**.
- 2 In the **Address Maintenance** dialog, click **Add** and the Address Properties dialog appears.
- 3 Enter the name for the notification in the Name text box.
- 4 **OPTIONAL** In **Description**, enter the description for this addressee.
- 5 Click **Add** and the Address Detail dialog appears.
- 6 Click the **Address Role** list, and then select the target role.
- 7 Click the **Address Type** list, and then select **Tivoli eMsg**.

8 Specify the details for the Tivoli eMsg. The table below describes the Embarcadero_Event slots (variables):

Slot	Contents	Information
ops_reqd	YES/NO	Defaults to NO
email_reqd	YES/NO	Defaults to NO
pager_reqd	YES/NO	Defaults to NO
ticket_reqd	YES/NO	Defaults to NO
email_to	STRING	List of recipients
email_sub	STRING	Subject for the message
email_msg	STRING	Actual message
ticket_short	STRING	Short description
ticket_details	STRING	Details of the event
pager_to	STRING	List of recipients
pager_msg	STRING	Message to display on pagers
opco	STRING	FPL, etc.
sub_source	STRING	Name of script generating this event
hostname	STRING	Name of server having the problem

NOTE: STRING slots can only hold a maximum of 255 characters and YES/NO must be upper case.

Postemsg Format

The following is an example of the postemsg format:

```
postemsg -f configuration file -r severity -m message [slot=value] class source
```

The table below describes the variables in the example:

Variable	Description/Example
configuration file	The directions for postemsg to get the event to Tivoli, supplied by the user.
severity	The severity of the postemsg -FATAL, CRITICAL, MINOR, WARNING, or HARMLESS. NOTE The severity must be upper case.
message	The text that will be displayed on the Tivoli Event Console (TEC).
slot=value	There is no order of importance in the pairing. NOTE An opco slot must be upper case.
class	The name of the event, e.g. Embarcadero_Event. NOTE The class is case sensitive.
source	This is used by TEC to assign the event to a configured source. For example, SENTRY is the source for monitoring. NOTE SENTRY must be upper case.

Examples of Events

The table below list some examples of events:

Event	Format
Event Requiring E-mail Notification	Postemsg -f /usra/Tivoli/data/aix027.conf -r CRITICAL -m "This is a sample of a message" email_reqd=YES email_to="tom@xyz.com sue@abc.com" email_sub="This a sample subject" email_msg="Any text message you want to send" opco=FPL sub_source=YourScriptName hostname=ServerName Embarcadero_Event SENTRY

Event	Format
Event Requiring Pager Notification	Postemsg -f /usra/Tivoli/data/aix027.conf -r MINOR -m "This is a sample of a message" pager_reqd=YES pager_to="tom@skytel.com sue@sprint.com" pager_msg="Message to pager" opco=FPL sub_source=YourScriptName hostname=ServerName Embarcadero_Event SENTRY
Event Requiring a Ticket	Postemsg -f /usra/Tivoli/data/aix027.conf -r WARNING -m "This is a sample of a message" ticket_reqd=YES ticket_short="A meaningful short message" ticket_details="The real meat of the problem" opco=FPL sub_source=YourScriptName hostname=ServerName Embarcadero_Event SENTRY
Event Requiring All Methods of Notification	Postemsg -f /usra/Tivoli/data/aix027.conf -r MINOR -m "This is a sample of a message" email_reqd=YES email_to="tom@xyz.com sue@abc.com" ticket_reqd=YES ticket_short="A meaningful short message" ticket_details="The real meat of the problem" email_sub="This a sample subject" ticket_reqd=YES ticket_short="A meaningful short message" email_msg="Any text message you want to send" pager_msg="Message to pager" opco=FPL pager_reqd=YES pager_to="tom@skytel.com sue@sprint.com" ticket_details="The real meat of the problem" sub_source=YourScriptName hostname=ServerName Embarcadero_Event SENTRY

- 9 Click **OK** to save the details.
- 10 Click **OK** to close the Address Maintenance dialog.

Creating and Testing the Tivoli Notification

- 1 Click **Maintenance > Notifications**.
- 2 In **Notification Maintenance** click **Add** to open the Notifications Properties dialog.
- 3 In **Name**, type the name for the notification.
- 4 In **Description**, type the description for the notification.
- 5 Add a recipient with a **Tivoli eMsg** type.
- 6 Click **Test** and a test message is sent to the target address.
- 7 Click **OK** to save the selection.
- 8 Click **OK** once again and the changes are saved.
- 9 Click **Close** to close the Notification Maintenance dialog.

If the Test was successful you are now ready for the next step, [Attaching the Tivoli Notification to an Alert](#).

Attaching the Tivoli Notification to an Alert

- 1 On the **Home** view or the **Memory** view, right-click the **Buffer Cache Hit Ratio** statistic and click **Alerts**.
- 2 In **Datasource Alert Maintenance**, select the critical alarm, and then click **Edit**.
- 3 In **Alert Properties**, click the **Generate Alarm if** list, and then click **is not equal to**.
- 4 In **Value**, change the value to **1**.
- 5 In **Available Notifications**, select the Tivoli eMsg notification, and click the arrow to move it to the **Selected Notifications** box.
- 6 Click **OK** to save the changes and return to the Datasource Alert Maintenance dialog.

- 7 Click **Close** and the Datasource Alert Maintenance dialog closes and at the specified interval, the notification is sent.

Attaching a Notification

Performance Center lets you attach one or more notifications to multiple thresholds. You can attach notifications to all or specific thresholds or globally to datasource thresholds. You can attach notifications from the following dialogs:

- [Datasource Template Maintenance](#)
- [Alert Template Maintenance](#)

To attach notifications, you must log on the target Performance Center Server as an [Administrator](#).

Attaching a Notification

- 1 Open one of the following maintenance dialogs:

[Datasource Template Maintenance](#)

[Alert Template Maintenance](#)

TIP: Use the Datasource Template to add a notification(s) to specific threshold violations.

- 2 In the target dialog, select the target template and click Click **Attach Notif**.
- 3 In the **Attach Notification** dialog, select the target notification(s).

NOTE: The Attach Notification dialog is populated with existing notifications. For information on creating notifications, see [Adding Notifications](#).

- 4 Click **Attach** and the notification(s) are attached to the target template.
- 5 Click **Close**.

Deleting a Notification

Performance Center lets you delete [notifications](#). To delete a notification, you must log on to the target Performance Center Server as an [Administrator](#).

Deleting a Notification

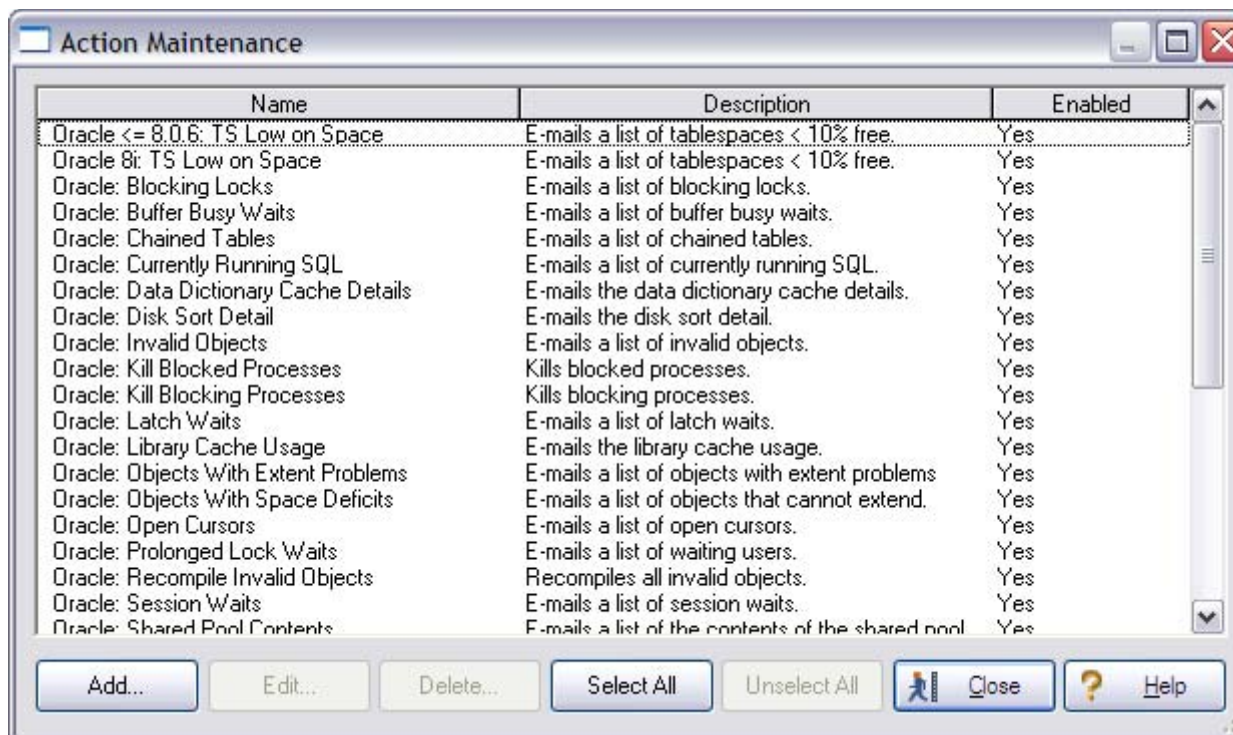
- 1 Click **Maintenance > Notifications**.
- 2 In **Notification Maintenance**, select the target notification
- 3 Click **Delete** and a confirmation message appears.
- 4 Click **Yes**. The confirmation message closes and the notification is deleted.

Actions

Actions are programs or scripts that you define to run in response to detected threshold violations, which occur in Performance Center. Performance Center provides an Action Maintenance dialog that lets you manage actions for all monitored datasources. Performance Center includes several sample actions that you can run against your monitored datasources.

The table below describes the properties available in the Action Maintenance dialog:

Option	Description
Name	Lets you see the name of each action.
Description	Displays the description of each action, if any.
Enabled	Lets you see if the action is enabled.



If you are logged on as an [Administrator](#), the following functionality is available for actions:

- [Add](#)
- [Edit](#)
- [Delete](#)

For more information, see:

[Using the Action Scripts Setup](#)

[Adding an Action](#)

[Editing an Action](#)

[Deleting an Action](#)

[Action Maintenance](#)

[Opening the Action Maintenance dialog](#)

Using the Action Scripts Setup

With the new action scripts setup, now you can create a new SQL action with a specified script from the Performance Center client. Actions that execute arbitrary commands are now easier to use. Rather than requiring that the action include the logic to send an e-mail/notify the user, the standard output is now captured from the command and attached to the action notification.

An [Action Script Example](#) follows.

NOTE: However, you can still able to execute a SQL script using the arbitrary command execution.

When you create a new Action, you have the option to configure three different types of Actions:

Action Type	Description
Command	<p>Command Text: Select existing executable (.exe or .bat) files from a Command File dialog.</p> <p>Arguments: Select an argument from a pull-right menu. The options available are: Datasource, Server, Statistic, Alert, and Environment. Your selection then appears in the Arguments list box.</p> <p>Initial Directory: Browse for and select an Initial Directory path. The full path appears in this field.</p> <p>Once you click Finish the new command appears in the Action List Box.</p>
SQL	<p>SQL Statement: Enter the SQL statement that you want to enact for the new action.</p> <p>Once you click Finish the new command appears in the Action List Box.</p>

Action Type	Description
DB Optimizer	<p>DBOptimizer Path: Opens the DBOptimizer executable file dialog where you can browse to the path where the DBPptimizer application resides.</p> <p>Output Folder: Opens the Browse for Folder dialog. Select the folder where you want the results to appear. The full path appears in this field.</p> <p>Workspace: Browse to the folder where the dboptimizer workspace resides.</p> <p>Test Button: Verifies configuration/privileges on the Performance Center server. In the Test Action dialog, select your datasource from the datasource box which lists all registered datasources. Click Run and the script that will be run appears in the text area.</p> <p>Time Interval: Set the time interval (in minutes) of how often you want this action to occur.</p> <p>Once you click Finish the new command appears in the Action List Box.</p>

DB Optimizer

This topic covers additional information on using Action Scripts and DBOptimizer.

Launching DBOptimizer doesn't involve any explicitly-specified command line arguments (the data source argument is inferred from the monitored data source).

Listed below are two examples of the command line options for DB Optimizer. The Profiling example is related to a performance alert where the DBA wants more information. The Tuning example shows a command line option in PC. This tuner command line option is more for tools like DB Artisan that might want to send a query to DBO.

Profiling:

```
dboptimizer.exe profile ds:<data source> duration:<profile session duration in minutes> tofile:<profile session save file.oar> <overwrite>
```

Example:

```
dboptimizer.exe profile ds:romlaborcl8i_1 duration:120 tofile:c:\testtest.oar
overwrite
```

NOTE: Duration was measured in seconds in DB Optimizer 1.5. Starting in DB Optimizer 1.5.1 it is measured in minutes.

Tuning:

```
dboptimizer.exe tune ds:<data source> sqlfile:<path to sql file>
```

Example:

```
dboptimizer.exe tune ds:romlaborcl8i_1 sqlfile:c:\sqlfile.sql
```

For more information, see: [Setting Up an Action Script to DBOptimizer](#) and [Tuning SQL with DB Optimizer](#).

Setting Up an Action Script to DBOptimizer

There are two requirements for using the DB Optimizer Command Line option:

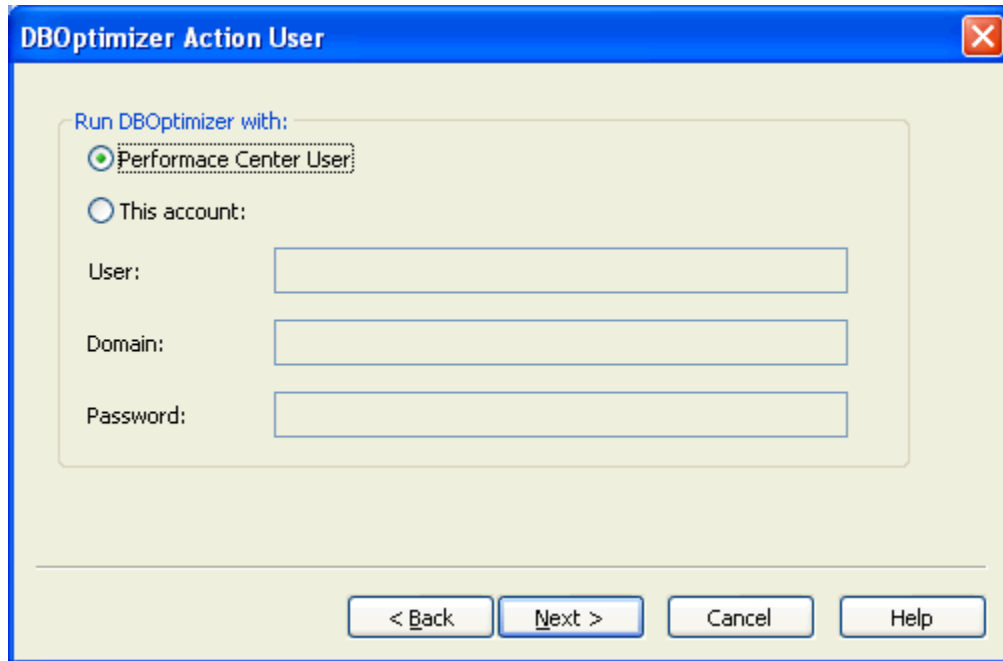
- The specified data source has to be already registered, with correct credentials.
- The files have to be correct. If the overwrite option is in the script, the file will be overwritten.

Each tool has its own set of data sources and it is recommend that the user keep the data source names the same on each tool.

- 1 From the main menu select **Maintenance > Actions** and the Action maintenance dialog appears.
- 2 Click **Add** to open the Action Properties dialog.
- 3 Enter the name of the action you want to create and click the Enabled option.
- 4 **Optional:** Enter the description of the action.
- 5 In the Action Type section select **DBOptimizer** and click **Next**.

The screenshot shows the 'Action Properties' dialog box. The 'General' section contains a 'Name:' text box, a 'Description:' text box, and a checked 'Enabled' checkbox. The 'Action Type' section contains three radio buttons: 'Command', 'SQL', and 'DBOptimizer', with 'DBOptimizer' selected. At the bottom, there are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

- 6 You have two options to run DBOptimizer with: **Performance Center User** and **This account**.



Performance Center User

If the Performance Center service runs as a local windows user who installed DB Optimizer, then it should be able, by default, to read and write to those users directories and no extra steps are necessary.

For example if I, user “user1”, installed DB Optimizer on my desktop then my workspace and datasource directories would be in

C:\documents and settings\user1

For example:

- **Workspace** (by default)
C:\documents and settings\user1\dboptimizer\workspace
- **Datasources** (by default)
C:\documents and settings\user1\Application Data\embarcadero\workspace
- **Metadata** (by default)
C:\documents and settings\user1\Application Data\embarcadero\workspace\metadata

This Account

If you select this option, you must supply custom credentials (user, domain, and password). If the “Embarcadero Performance Center” service is running as “Local System” the files have to be located outside of the local user’s directory.

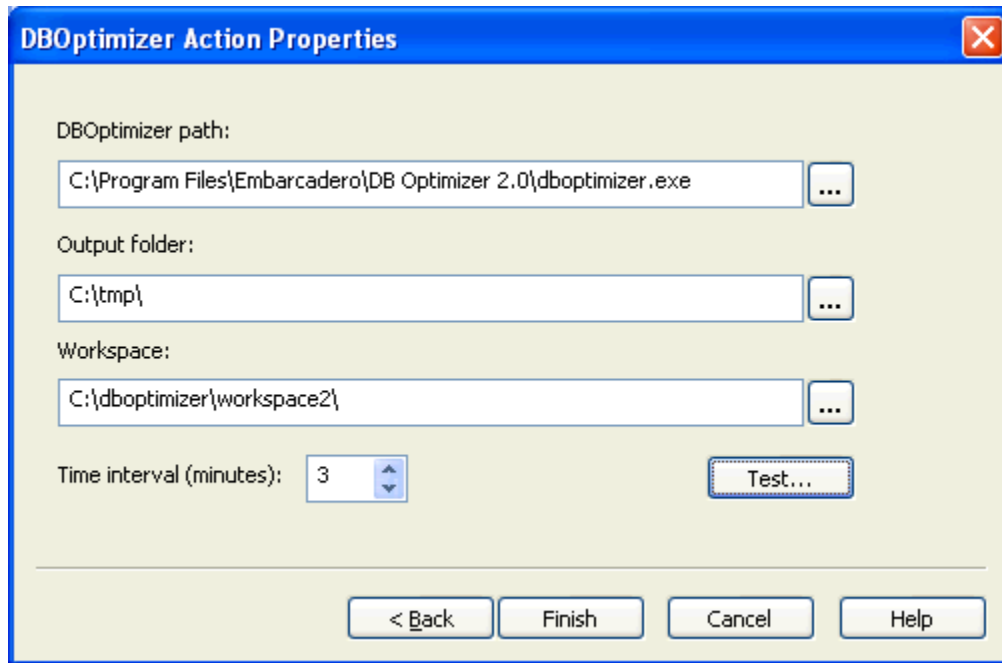
The Performance Center service user, “Local System” can’t access these user’s directories, so these directories have to be moved to a location such as:

C:\dboptimizer

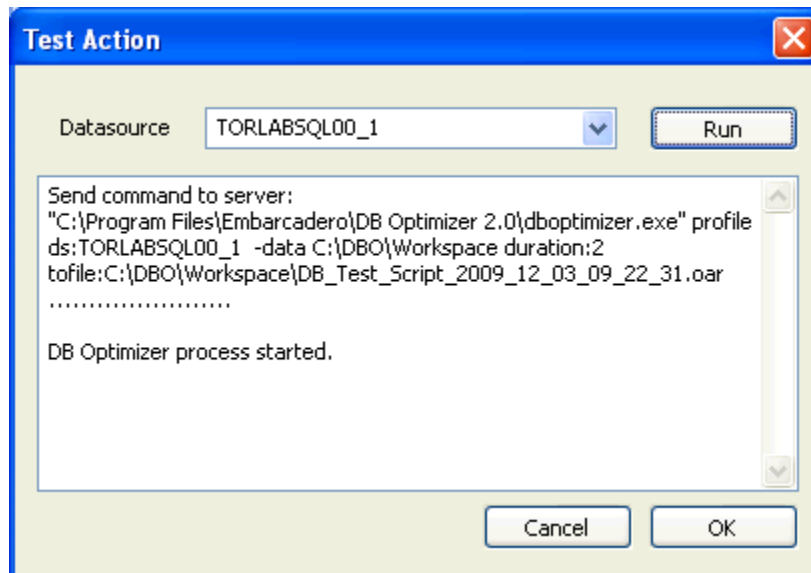
For example:

- C:\dboptimizer\workspace (the workspace used for profiling)
- C:\dboptimizer\outputfiles\ (the folder that will contain the output of the profiling session. The name of the file will be in the form [action_name][yyyy_MM_dd_hh_mm_ss].oar.

- After you have populated the text fields in the DBOptimizer Action Properties dialog, click **Test**. The Test Action dialog appears.



- Select the Datasource from the drop-down list and click **Run**. The Performance Center server will start a profiling session with a duration of two minutes on the target data source.



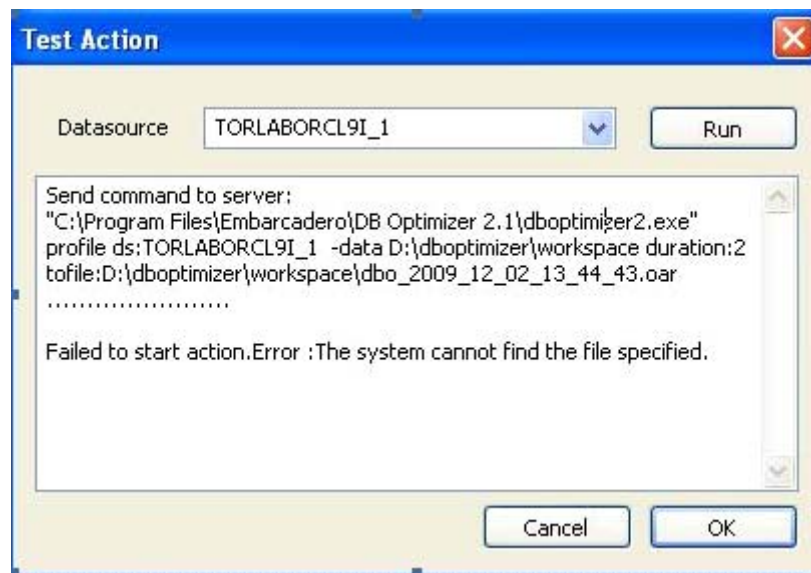
The test action output will look like the screen shot shown above if the action is started. In this case you will have to check if the output file is created.

NOTE: It takes some time until the oar file will appear in output folder because DBO has to start (approximately 1-2 minutes) and after that it will take another 2 minutes for profiling

To accomplish this, the Performance Center client will send a command to the server to execute the action and the command line will be shown in the output window:

- The command sent to server is based on the properties entered in the Properties dialog.
- DBOptimizer will use a workspace in the folder with the same name as the target datasource that is located in the folder previously selected.
- You should check the output folder to see if the file [output_folder]\[action_name_yyyy_mm_dd_hh_mm_ss].oar was created. If this file wasn't created, you can check the logs from DBOptimizer that should be in the workspace folder.

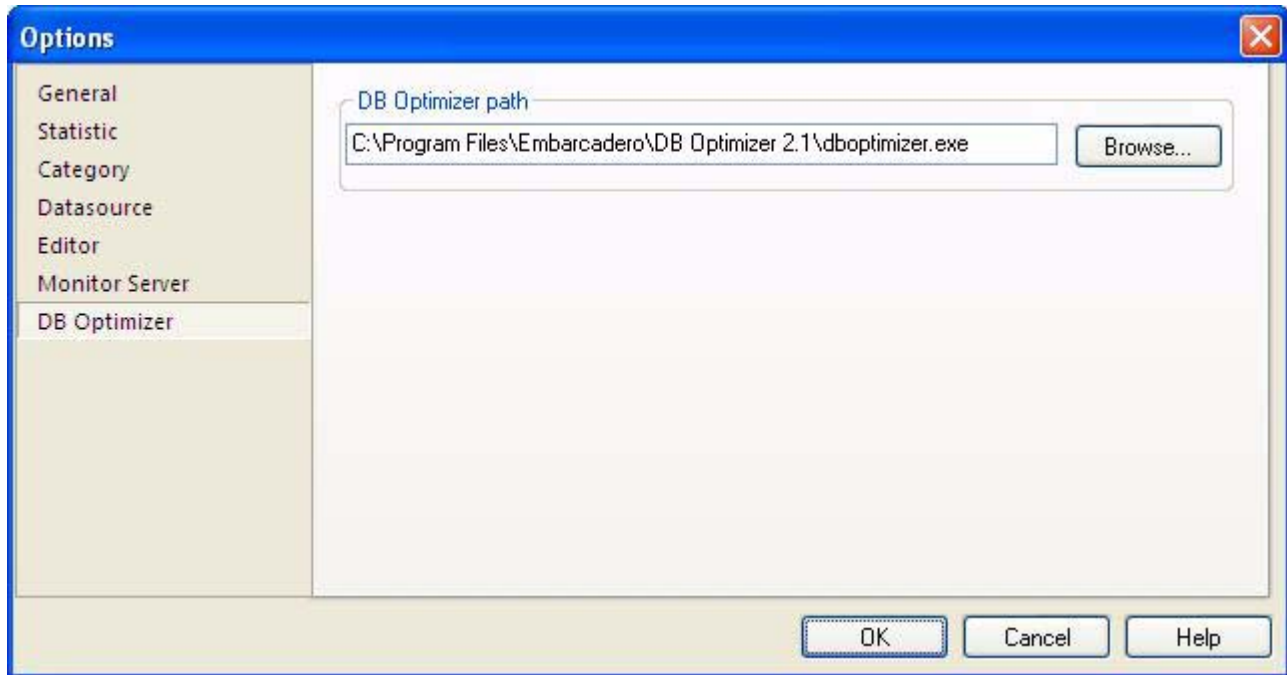
NOTE: In Case of an error the output could resemble the following screen shot. In this example the path to the DB Optimizer executable was incorrect.



Tuning SQL with DB Optimizer

Tune SQL is now executed with DB Optimizer.

- 1 In the Datasources pane, double-click **Top SQL** and the Top SQL dialog appears.
- 2 Select your criteria and click **Show SQL** (located in the upper right-hand corner of the dialog).
- 3 Choose a 'SELECT' query from the SQL Text list and click **Tune SQL**. DB Optimizer starts a tune session for the selected query. If the path to DB Optimizer was not set in the client option previously, the Client will ask for you for the path



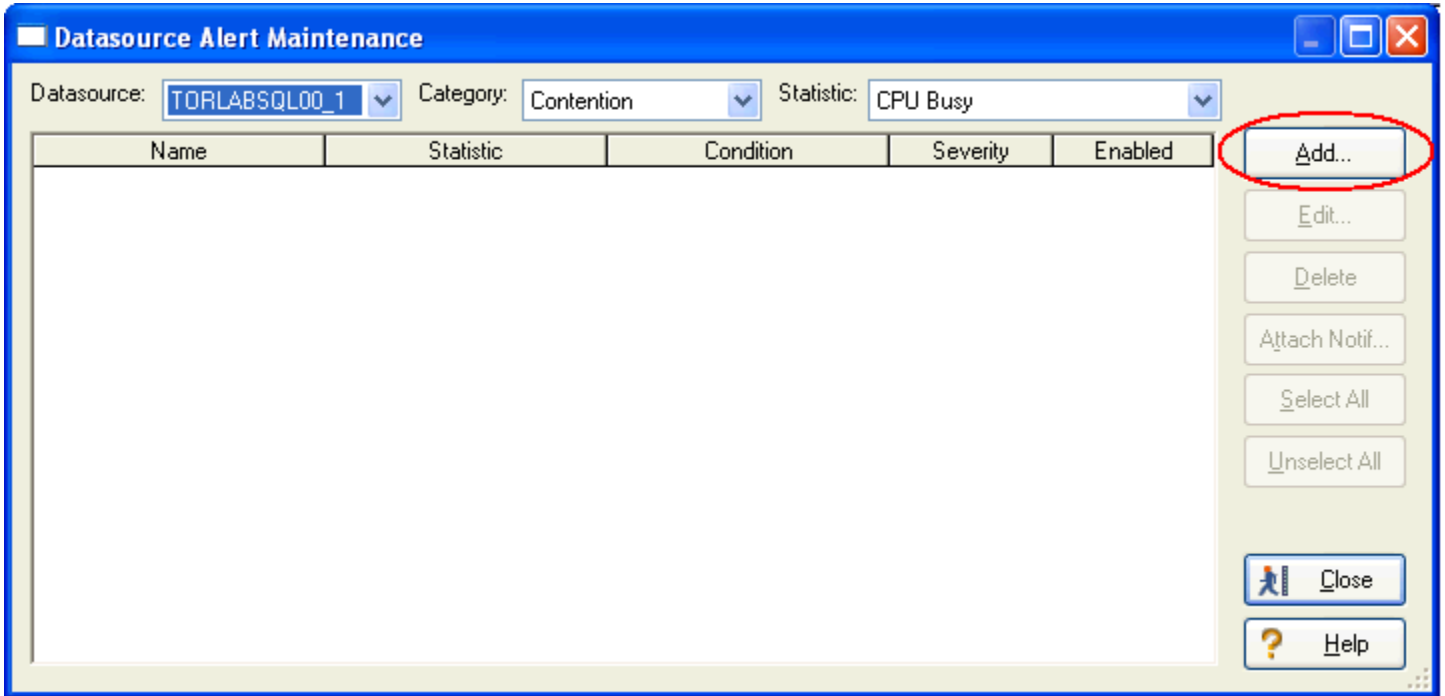
Action Script Example

The following is a brief walkthrough of how to get information about active sessions. In this example we will set a SQL script to get an e-mail when CPU usage on your server gets too high.

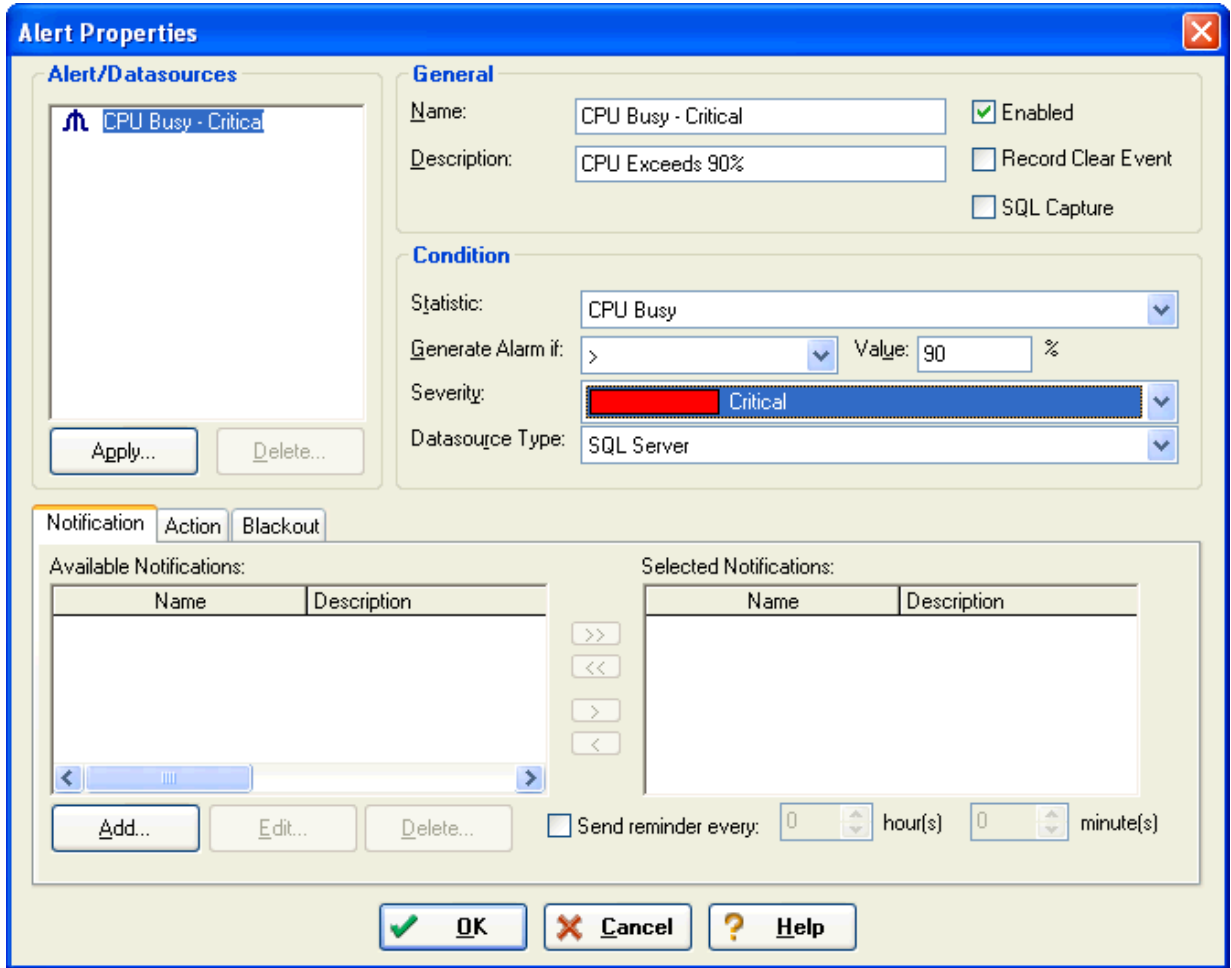
- 1 In your registered datasource, open the Contention tab.
- 2 Right-click on CPU Busy, and click **Alerts** on the context menu.

The screenshot displays the 'TORLABSQL00_1 - CONTENTION' dashboard. At the top, a 'Contention Health Index' bar is at 100%. Below it are navigation tabs: HOME, MEMORY, I/O, SPACE, DATABASES, CONTENTION (selected), and USERS. The dashboard shows 'System Contention' metrics: Users Blocked (0), Page Deadlocks (0.00 /s), Latch Waits (0.00 /s), Log Flush Waits (0.04 /s), and Table Lock Escalations (0.00 /s). A line graph shows Log Flush Waits over time. To the right is a 'Lock Overview' table with one entry: 'Waiting for Data'. Below the graph are 'Cpu Busy' (0.42%), 'IO Busy' (0.33%), and 'Server Idle' (99%) sections. A context menu is open over the 'Alerts...' button in the Cpu Busy section, with options: Alerts..., Properties..., and Help. The bottom of the window shows a message log with four entries and a taskbar with 'localhost: Started'.

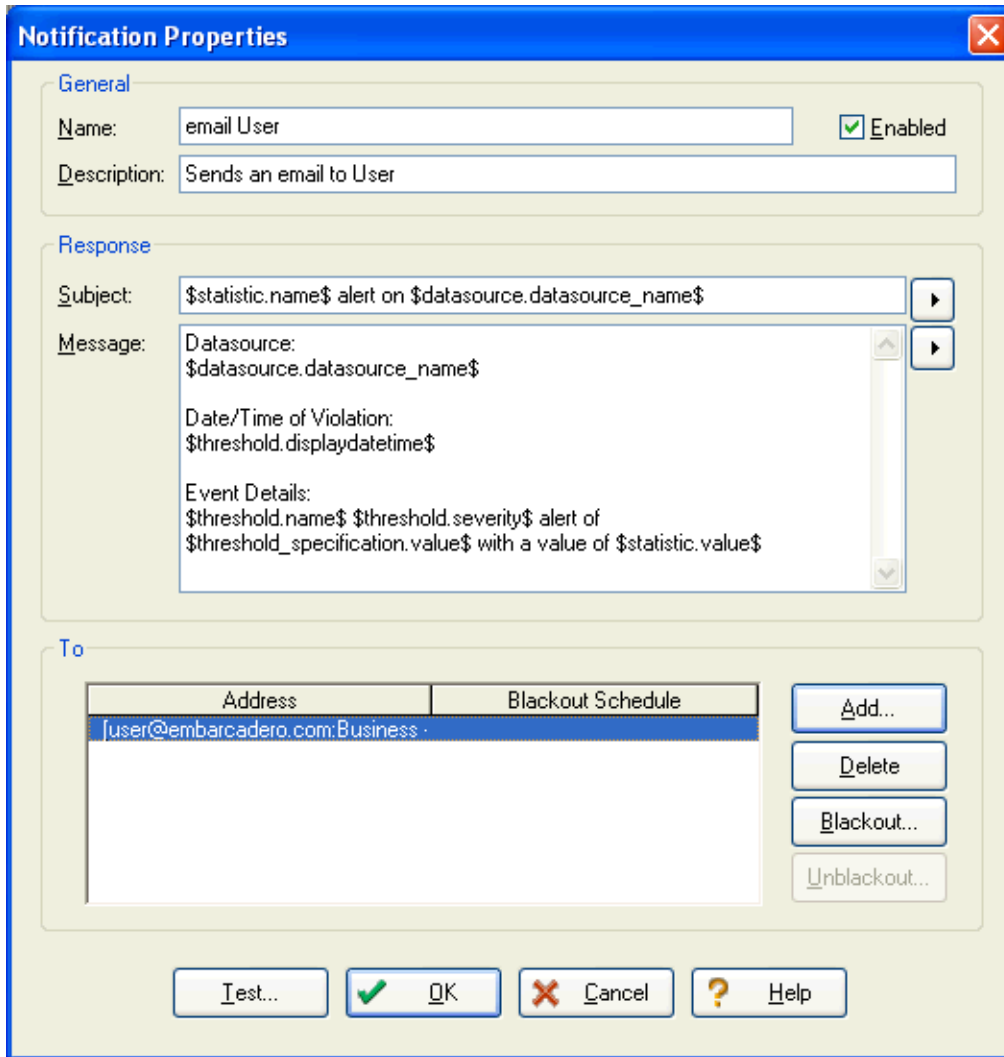
- 3 In the Datasource Alert Maintenance dialog, click **Add** to add a new alert.



- 4 In the Alert Properties dialog, enter the appropriate descriptions and values you want to include.

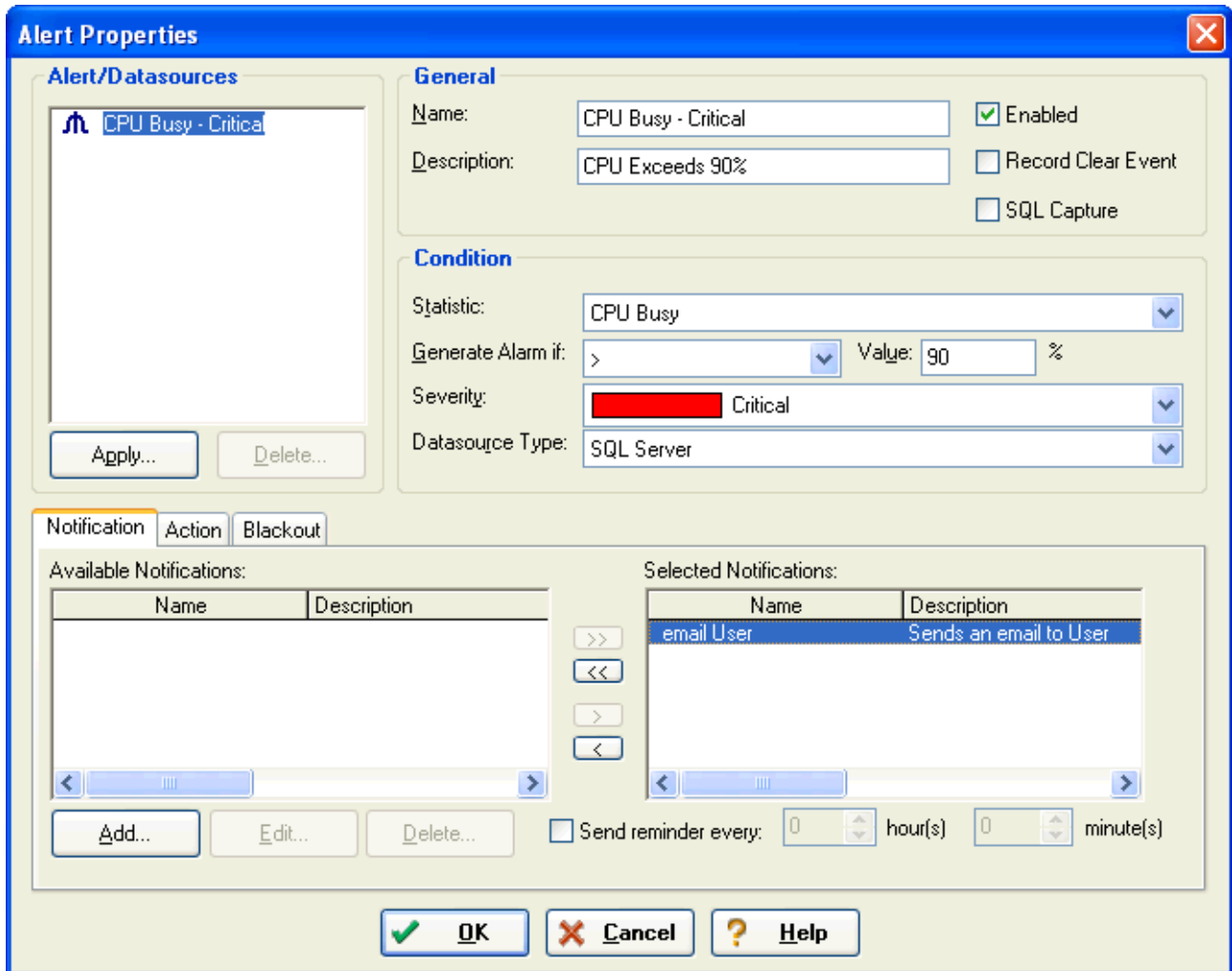


- 5 Under the Notification tab, click **Add** to open the Notification Properties dialog and configure the appropriate notification.

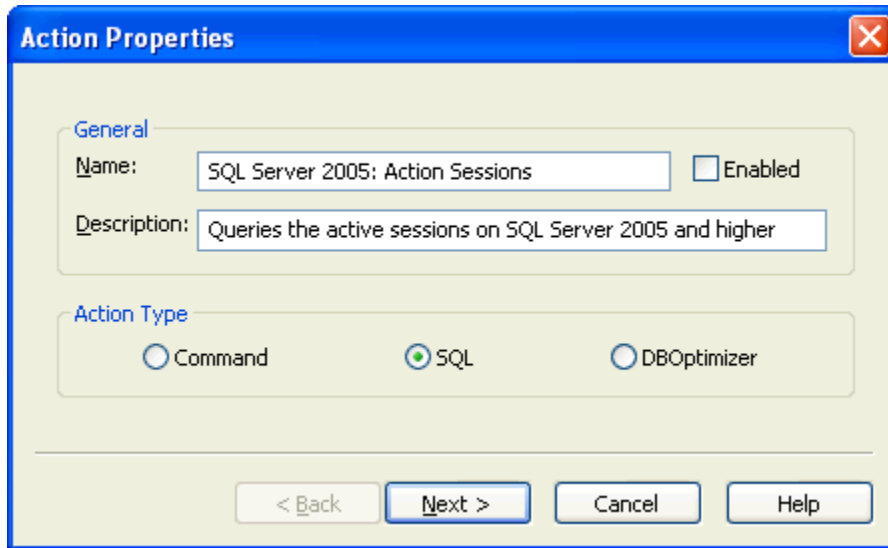


NOTE: You can also add a e-mail address in this dialog.

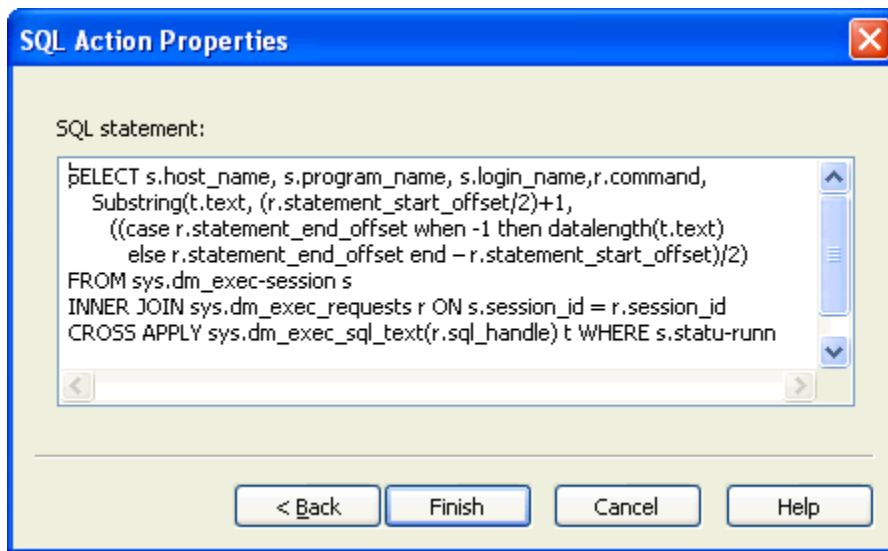
- Once you have established a Notification, in the Alert Properties dialog select that new notification and move it to the Selected Notifications list box.



- 7 Click the Action tab and then click **Add**. The Action Properties dialog appears. Enter the name and description of the action. For this example, select SQL. However if you want to [execute](#) a command on the Performance Center Server or launch DBOptimizer, select those options instead.

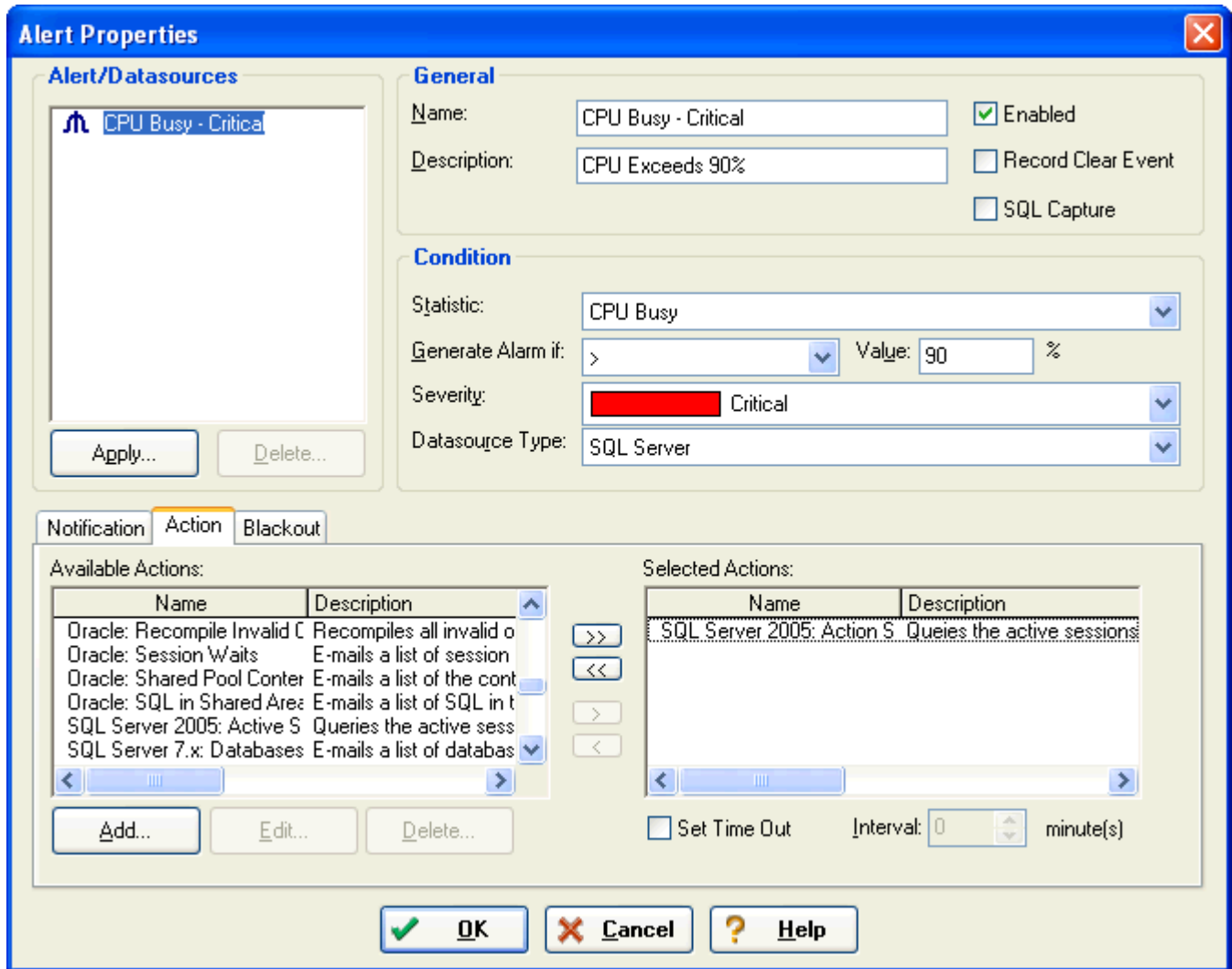


8 Once you select SQL, click **Next** to open the SQL Action Properties dialog.



9 Enter the SQL statement and click **Finish**. The newly created Action appears in the Action list box.

10 Select this new action in the list box and move it to the Selected Action list box.



11 When the alert fires, you will receive the result of the SQL action as part of the notification e-mail.

NOTE: If you are using Outlook, you may get the message “Extra line breaks in this message were removed”. Click on that to put the line breaks back in (otherwise all your rows will blend together).

Datasource:
ROMLABPCSQL05

Date/Time of Violation:
Tuesday, July 14, 2009 13:28:46

Event Details:
CPU Busy - Critical Critical alert of 5 with a value of 13.572401271009

The following action was run:

Action name: SQL Server 2005: Active Sessions
Action output:

```
host_name, program_name, login_name, command, text
ROWMFROH01, DBArtisan, sa, SELECT, SELECT * FROM sys.all_objects a , sys.all_objects b ,
      sys.all_objects c, sys.all_objects d
ROVMMFROH05, DataServer Module, sa, SELECT, SELECT s.host_name, s.program_name,
      s.login_name,r.command,
      substring(t.text, (r.statement_start_offset/2)+1 ,
      ((case r.statement_end_offset when -1 then datalength(t.text)
      else r.statement_end_offset end - r.statement_start_offset)/2) + 1)
      as text
FROM sys.dm_exec_sessions s
INNER JOIN sys.dm_exec_requests r ON s.session_id =session_id
CROSS APPLY sys.dm_exec_sql_text(r.sql_handle) t WHERE s.status=unning'
```

Opening the Action Maintenance dialog

- Click **Maintenance > Actions** and the Action Maintenance dialog appears.

Adding or Editing an Action

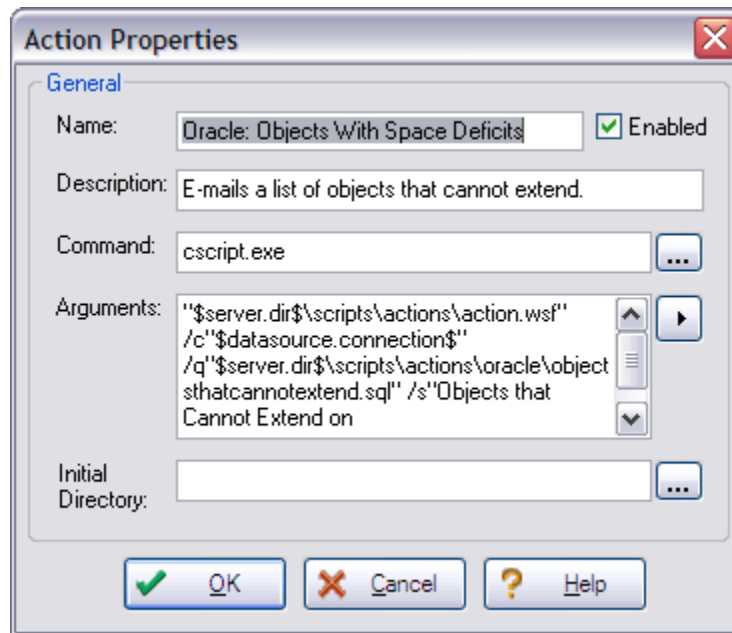
These are programs or scripts that you define to run in response to detected threshold violations.

You can add [actions](#) to the repository, and it lets you edit addresses in the repository. You must be logged into the target Performance Center Server as an [Administrator](#) to add or edit actions.

You add or edit actions using the Action Properties dialog. The table below describes the options and functionality of the Action Properties dialog:

Option	Description
Name:	Type or change the action's name.
Description:	OPTIONAL Type or change the action description.
Command:	Type or browse for the action's executable or target file.

Option	Description
Arguments:	OPTIONAL Type or edit arguments to pass to the command. You can also populate the box with the target values by clicking the arrow next to the Arguments box. NOTE Arguments are case-sensitive.
Initial Directory:	OPTIONAL Enter the path of the file directory for the action.
Enabled	Enable the action.



For more information, see

[Opening the Action Properties dialog](#)

[Adding or Editing an Action](#)

Opening the Action Properties dialog

- 1 Click **Maintenance > Actions**.
- 2 In the **Action Maintenance** dialog, click **Add** or **Edit** and the Action Properties dialog appears.

Adding or Editing an Action

- 1 Click **Maintenance > Actions**.
- 2 In the **Action Maintenance** dialog, click **Add** or **Edit** to open the Action Properties dialog.
- 3 In **Name**, enter the name for the action.
- 4 **OPTIONAL** In the **Description** box, type the description for the action.
- 5 In **Command**, enter the target file or executable for the action.

- OPTIONAL** In **Arguments**, type arguments to pass to the command or click the arrow next to the **Arguments** box to populate the box with the target values.

NOTE: Arguments are case-sensitive.

- OPTIONAL** In **Initial Directory**, enter the path of the file directory for the action.
- To enable the action on completion of these steps, select the **Enabled** option and click **OK**.

Deleting an Action

Performance Center lets you delete [actions](#). You must be logged into the target Performance Center Server as an [Administrator](#) to delete actions.

Deleting an Action

- Click **Maintenance > Actions**.
- In **Action Maintenance**, select the target action.
- Click **Delete** and a confirmation message appears.
- Click **Yes** and the confirmation closes and the action is deleted.

Schedules

Schedules are time intervals where Performance Center will not send notifications about any normal threshold violations or performance problems. For example, if you know you will run an Oracle database cold backup every day between three and four am, you can use the Schedule Maintenance to create a blackout schedule so that you are not notified. After you create schedules, you can use any of the Schedule dialogs to attach blackout schedules.

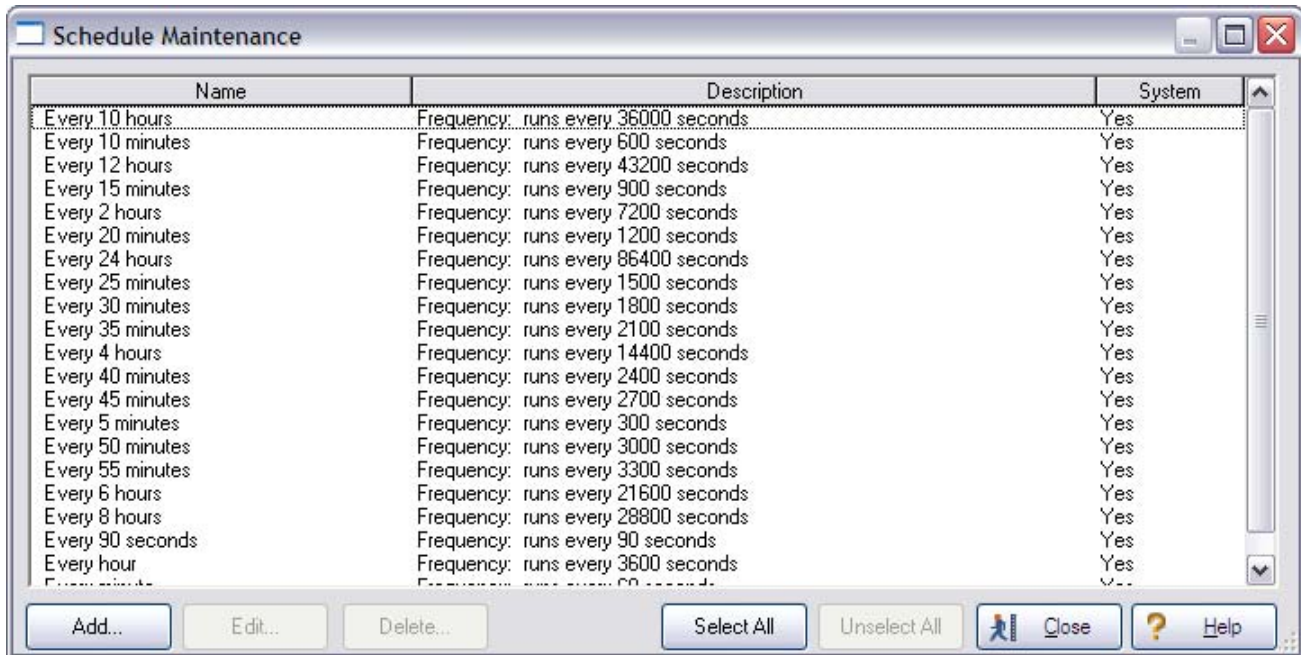
Performance Center uses the Microsoft Task Scheduler component of IE for scheduling some reports. You start the Scheduler service to schedule Performance Center reports.

Performance Center includes two dialogs to let you work with schedules. The table below describes these dialogs:

dialog	Description
Schedule Maintenance	View all blackout schedules stored in the Repository.
Select Schedule dialog	Select a schedule to run.

The table below describes the properties available in the Action Maintenance dialog:

Option	Description
Name	Displays the name of each schedule.
Description	OPTIONAL Displays the name of each schedule.



Performance Center Server offers the following functionality for schedules:

- [Add](#)
- [Edit](#)
- [Attach](#)
- [Detach](#)
- [Delete](#)

For more information, see:

[Adding a Blackout Schedule](#)

[Editing a Blackout Schedule](#)

[Attaching a Blackout Schedule](#)

[Detaching a Blackout Schedule](#)

[Deleting a Blackout Schedule](#)

[Select Schedule dialog](#)

[Schedule Maintenance](#)

[Opening the Schedule Maintenance dialog](#)

Opening the Schedule Maintenance dialog

- Click **Maintenance > Schedules**.

Adding or Editing a Blackout Schedule

You can add [blackout schedules](#) to the repository, and edit schedules stored in the repository. You must be logged into the target Performance Center Server as an [Administrator](#) to add or edit schedules.

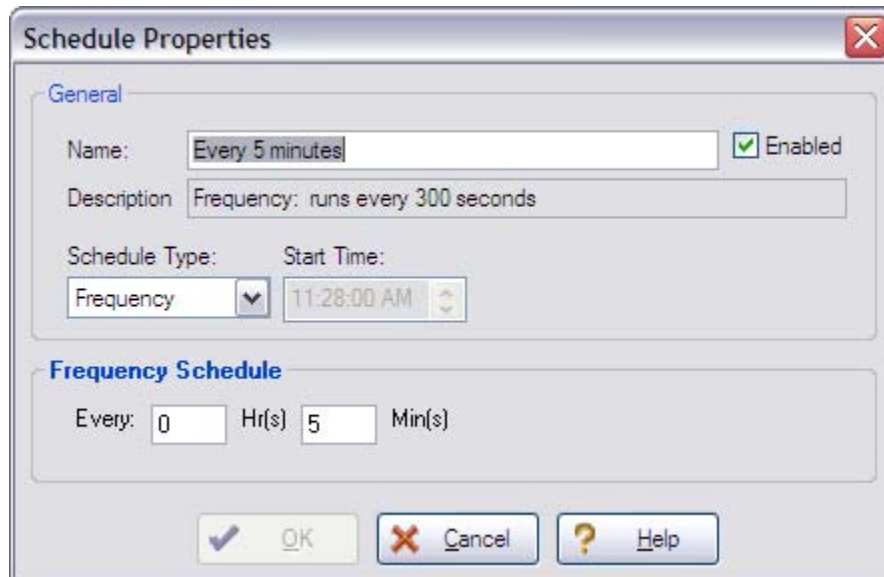
You can add or edit a schedule through one of the following dialogs:

- [Schedule Properties dialog](#)
- [Alerts Properties dialog](#)

Schedule Properties dialog

The table below describes the options and functionality of the Schedule Properties dialog:

Option	Functionality
Name	Type the schedule's name.
Enabled	Enable the schedule. It is selected as default.
Description	Displays the schedule parameters after you have made your selections.
Schedule Type	Choose schedule the type of schedule. Type or select start and end times.
Advanced	Opens the Advanced Schedule Options dialog.
Daily/Weekly/Monthly/Frequency Schedule	Add schedule details depending on the Schedule Type you selected. Daily - Displays that the schedule will run once each day. Weekly - Select the day(s) of the week on which to run the schedule. Monthly - Choose the month(s) and either the day of the month the rate and day(s) of occurrence. Frequency - Choose the frequency to run the schedule. you must specify the time.



For more information, see:

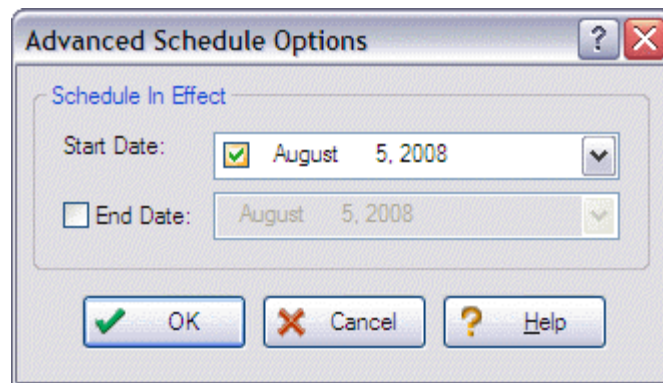
[Completing the Schedule Properties dialog](#)

[Advanced Schedule Options dialog](#)

Advanced Schedule Options dialog

The table below describes the options and functionality of the Advanced Schedule Options dialog:

Option	Description
Schedule in Effect Start Date	Select the start date for the schedule. This defaults to the current date.
Schedule in Effect End Date	Select the end date for the schedule.



For more information, see [Opening the Advanced Schedule Options dialog](#).

Opening the Advanced Schedule Options dialog

- 1 Click **Maintenance > Schedules**.
- 2 Click **Add** or **Edit** and the Schedule Properties dialog appears.
- 3 To set advanced scheduling options, click **Advanced** and the Advanced Schedule Options dialog appears.

Completing the Schedule Properties dialog

- 1 Click **Maintenance > Schedules**.
- 2 Click **Add** or **Edit** and the Schedule Properties dialog appears.
 - NOTE:** You can also open the Schedule Properties dialog from the Blackout tab of the Alerts Properties dialog.
- 3 In **Name**, type the name for the schedule.
- 4 To enable the schedule on completing these steps, select the **Enable** option.
- 5 After you complete the schedule, in **Description**, a description for the schedule appears.
- 6 Click the **Schedule Job** list, and then select the frequency for the schedule.
- 7 In **Start time**, enter the time to start the schedule.
- 8 To schedule the job to run once, click the **Run on** list, and then click the target run time.

- 9 Do one of the following:

To create the new schedule, click **OK**. A new schedule is created, the Schedule Properties dialog is closed, and you are returned to the Schedule Maintenance dialog.

OR

To set advanced scheduling options, click **Advanced** and the Advanced Schedule Options dialog appears.

- 10 In the **Advanced Schedule Options** dialog:

- To set the time to start the schedule, in **Start Date**, enter the target date.
- To set the end date, select the **End Date** option, and then enter the target date in the box.

NOTE: The available options in the Advanced Schedule Options dialog depend on the options you select in the Schedule Properties dialog.

- 11 Click **OK**. The Advanced Schedule Options dialog closes and returns to the Schedule Properties dialog.
- 12 Repeat [step 9](#).

Adding or Editing a Blackout Schedule for Alerts

- 1 [Open](#) the Alert Properties dialog.

If you are adding a threshold to which you will attach a schedule, complete the [Alert Properties](#) dialog, and then proceed to step 2.

If you are adding or editing a schedule to attach an existing threshold, proceed to step 2.

- 2 Click **Blackout**. The Blackout tab of the Alert Properties dialog appears.

To add a schedule, click **Add** and the Schedule Properties dialog appears.

OR

To view and/or modify a schedule, click **Edit** and the Schedule Properties dialog appears.

- 3 [Complete](#) the Schedule Properties dialog.
- 4 When you finish adding or editing the target schedule, click **OK**. The Schedule Properties dialog closes and displays the schedule in the Available Blackout Schedules dialog.
- 5 Use the arrows to move schedules to/from **Available Blackout Schedules**. The schedule is moved to the Selected Blackout Schedules box.

TIP: To move all the schedules, click the double arrows.

- 6 When you finish moving schedules, click **OK**. The schedule is attached, the Alert Properties dialog closes, and then returns to the Datasource Alert Maintenance dialog.

NOTE: If you are adding a new alert, Performance Center prompts you to add the alert to the currently selected datasource.

Attaching a Blackout Schedule

You can attach a [blackout schedule](#) to your [Notifications](#) and/or [Alerts](#). Attaching a schedule lets you set times when the application should not notify you about normal threshold violations or performance problems. For example, if you do not want to receive a notification about a regularly occurring backup, you can attach a blackout schedule to prevent Performance Center from sending notifications during this time. You can attach blackout schedules to:

- [Notifications](#)
- [Alerts](#)

For more information, see:

[Attaching a Blackout Schedule to a Notification](#)

[Attaching a Blackout Schedule to an Alert](#)

[Schedule Maintenance](#)

[Notification Properties](#)

[Alert Properties](#)

Attaching a Blackout Schedule to a Notification

You can use the [Notification Properties](#) dialog to attach a blackout schedule to an existing notification or to create a notification and attach a blackout schedule to it.

- 1 [Open](#) the **Notification Properties** dialog.
- 2 In the **To** box, select the target address.
- 3 Click **Blackout** and the Select Notification Schedule dialog appears.

TIP: The Select Notification Schedule dialog is populated with all of the schedules on the target Performance Center Server.

- 4 In the **Select Notification Schedule** dialog, select the target schedule, and then click **Attach**. The Select Notification Schedule dialog closes, returns to the Notification Properties dialog, and adds the schedule name to the Blackout Schedule column.

If you are adding a notification with a blackout schedule, proceed to step 5.

If you are attaching a blackout schedule to an existing notification, proceed to step 6.

- 5 [Complete](#) the **Notification Properties** dialog.
- 6 Click **OK** and the Notification Properties dialog closes.

Attaching a Blackout Schedule to an Alert

You can use the [Alerts Properties](#) dialog to attach a blackout schedule to an existing threshold or to create a threshold and attach a blackout schedule to it.

- 1 [Open](#) the **Alerts Properties** dialog.

If you are creating a new threshold with a blackout schedule, complete the [Alerts Properties](#) dialog, and then proceed to step 2.

If you are attaching a schedule to an existing threshold, proceed to step 2.

- 2 Click the **Blackout** tab and the Blackout tab of the Alerts Properties dialog opens.
- 3 Use the arrows to move the target schedules to **Available Blackout Schedules**. The schedule is moved to the Selected Blackout Schedules box.

TIP: To move all the schedules, click the double arrows.

- 4 When you finish moving schedules, click **OK**. The schedule is attached, the Alerts Properties dialog closes, and then returns to the Datasource Alerts Maintenance dialog.

NOTE: If you are adding a new threshold, Performance Center prompts you to add the threshold to the currently selected datasource.

NOTE: You can also add, edit, or delete a schedule when adding or editing a threshold.

Detaching a Blackout Schedule

You can detach a [blackout schedule](#) from a [notification](#). Detaching a schedule lets the Performance Center send a notification regarding threshold violations or performance problems.

For more information, see:

[Schedule Maintenance](#)

[Notification Properties](#)

[Alerts Properties](#)

Detaching a Blackout Schedule

- 1 [Open](#) the **Notification Properties** dialog.
- 2 In the **To** box, select the target address.
- 3 Click **Unblackout**. The blackout schedule detaches from the target notification and opens the Select Notification Schedule dialog.
- 4 [Complete](#) the **Notification Properties** dialog.
- 5 Click **OK** and the Notification Properties dialog closes.

Deleting a Blackout Schedule

Performance Center lets you delete blackout schedules. You must be logged into the target Performance Center Server as an [Administrator](#) to delete blackout schedules.

You can delete a blackout schedule through one of the following dialogs:

- [Schedule Properties](#)
- [Alerts Properties](#)

For more information, see:

[Deleting a Blackout Schedule from the Schedule Maintenance dialog](#)

[Deleting a Blackout Schedule from the Alerts Properties dialog](#)

Deleting a Blackout Schedule from the Schedule Maintenance dialog

- 1 Click **Maintenance > Schedules**.
- 2 In the **Schedule Maintenance** dialog, select the target schedule.
- 3 Click **Delete** and a confirmation message box appears.

- 4 In the message box, click **Yes** and the confirmation closes and the schedule is deleted.

Deleting a Blackout Schedule from the Alerts Properties dialog

- 1 [Open](#) the **Alert Properties** dialog.

If you are adding a threshold with a blackout schedule, complete the [Alerts Properties](#) dialog, and then proceed to step 2.

If you are attaching a blackout schedule to an existing threshold, proceed to step 2.

- 2 Click **Blackout**.
- 3 In the **Available Blackout Schedules**, select the target schedule, and then click **Delete**.
- 4 In the confirmation message, click **Yes**. The confirmation closes and the schedule is deleted.
- 5 When you finish deleting blackout schedules, click **OK**. The schedule is deleted, the Alerts Properties dialog closes, and then returns to the Datasource Alerts Maintenance dialog.

SQL Capture

SQL Text capture is a pro-active root-cause analysis tool and is now available in a 24/7 environment. It is an alert-based trackable instance of events. The information captured is stored in the repository for future analysis. The SQL Capture is triggered by a threshold violation. When such a violation occurs, a SQL Capture session is triggered and collects performance-related data for the SQL running currently on the datasource that fired the threshold.

You can set up how the notification takes place. As long as an alert is open you will receive notification based on the interval you set up.

For more information, see:

[Setting Up the Session Length](#)

[Setting Up the Capture Interval](#)

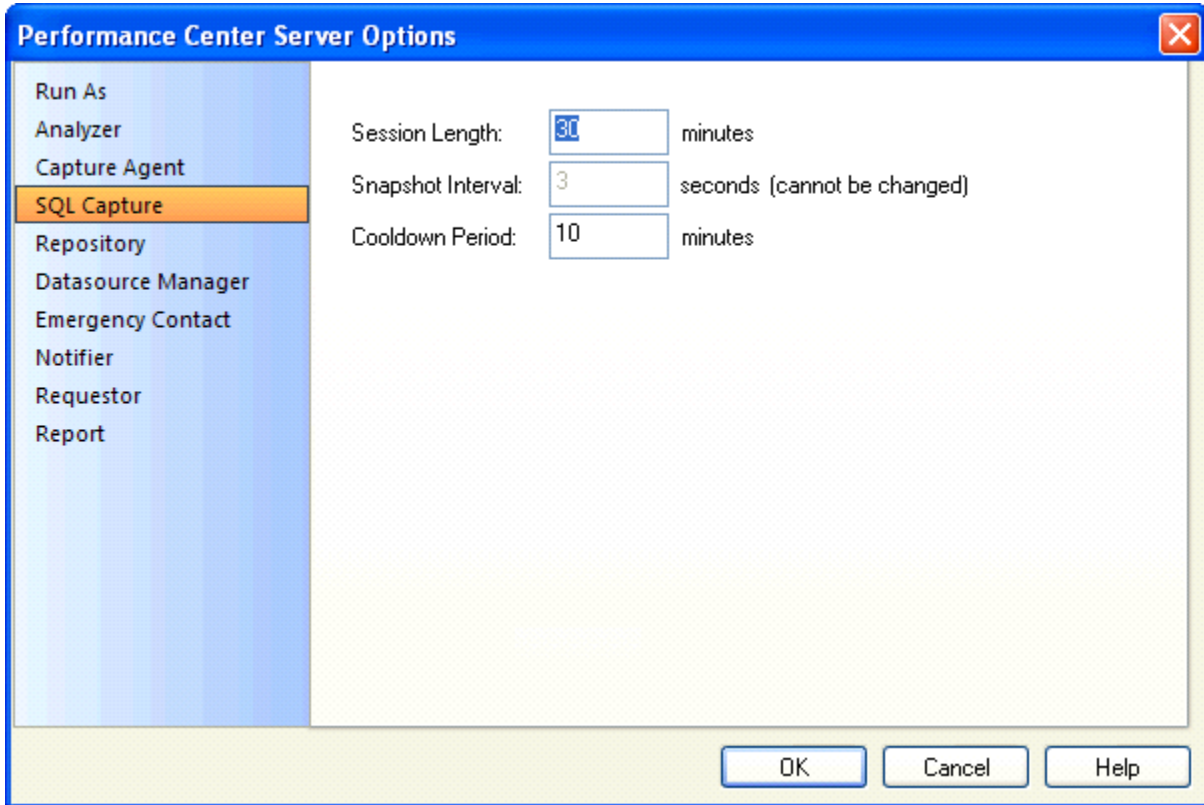
[Setting a SQL Capture Session on an Individual Alert:](#)

[Gaining Access to the Captured Information](#)

[Enable SQL Capture for Sybase ASE](#)

Setting Up the Session Length

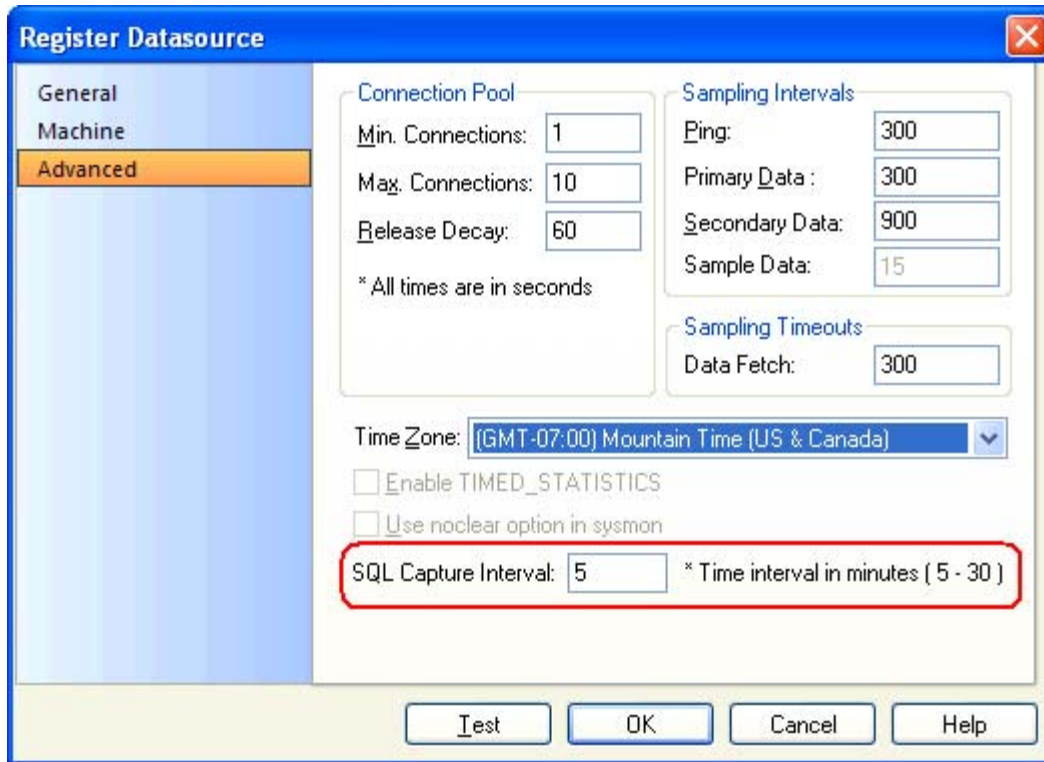
- 1 Select Monitor > Server Options.
- 2 Click SQL Capture.
- 3 Set the SQL capture interval. The default is 30 minutes.



- 4 You can also set the Cool down period. This setting determines how long to wait between consecutive capture sessions on a given data source. When an alert is triggered, if there is no existing SQL Capture Session on that data source, Performance Center checks to see when the last session ended. A new SQL Capture Session will not be started if the previous session's cool down period is not over.

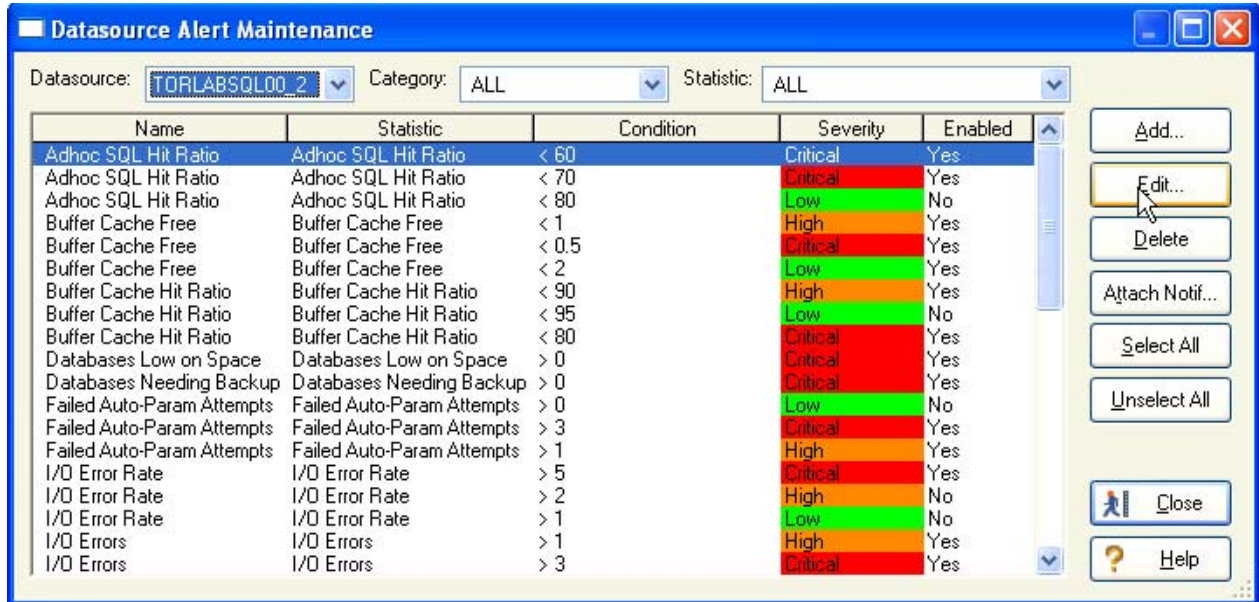
Setting Up the Capture Interval

- 1 In the Datasources tree, right-click the registered datasource you want to set the interval for and click **Properties**. In the Register Datasource dialog, click **Advanced**.
- 2 Set the interval time and click **OK**. The default is 30 minutes.

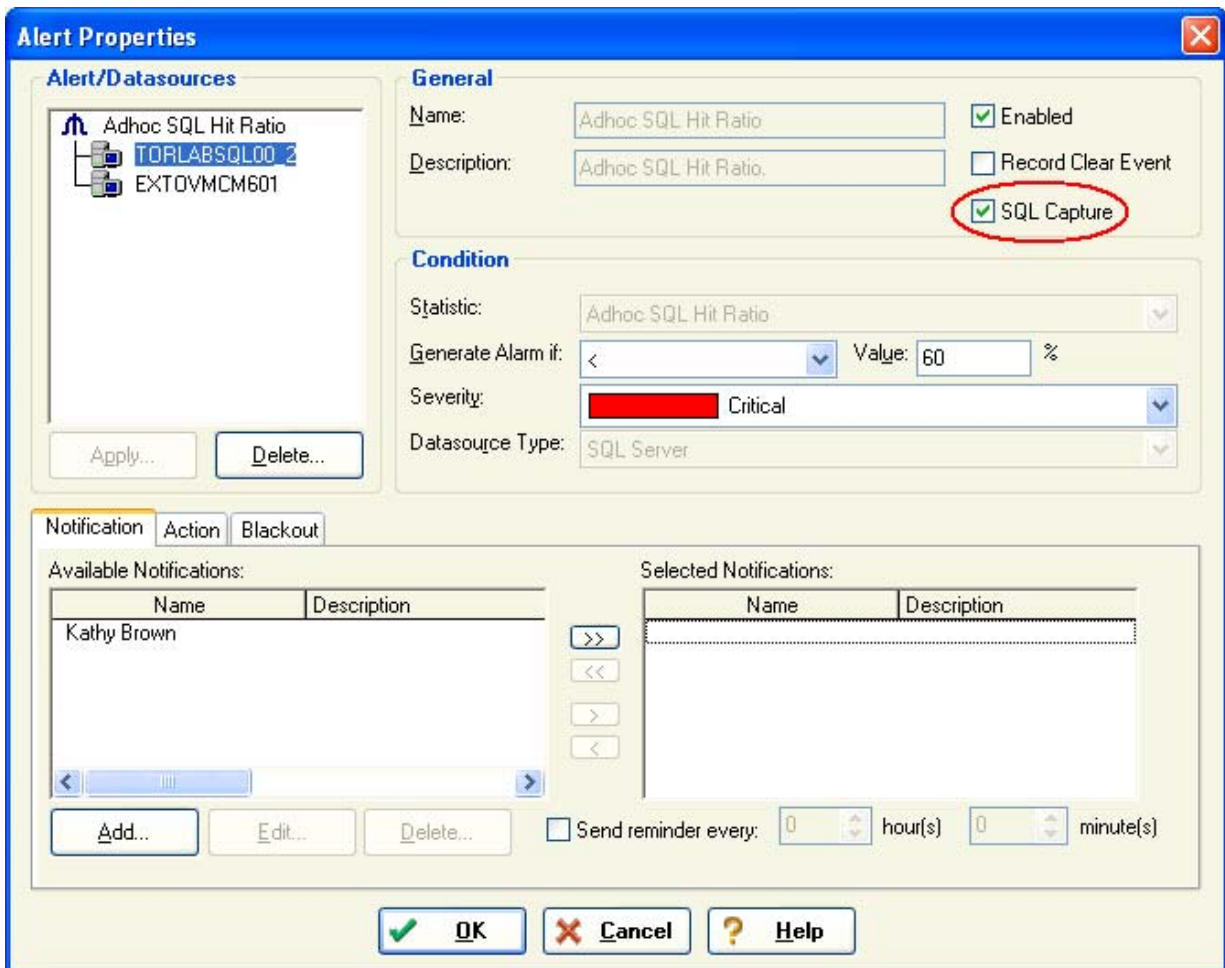


Setting a SQL Capture Session on an Individual Alert:

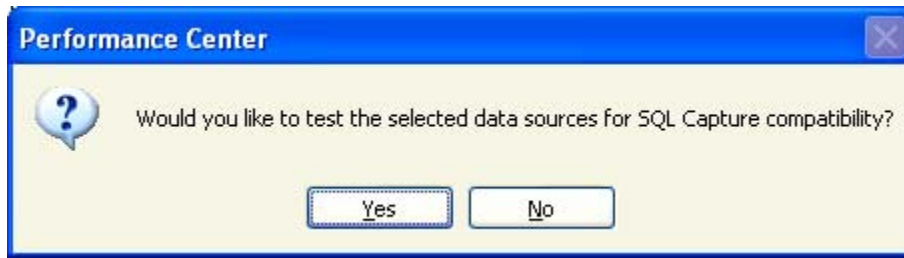
- 1 Click **Datasource > Alert Maintenance**.
- 2 In the Datasource Alert Maintenance dialog right-click the statistic you want to monitor and click **Edit**.



3 In the Alert Properties dialog select the SQL Capture option and click **OK**.



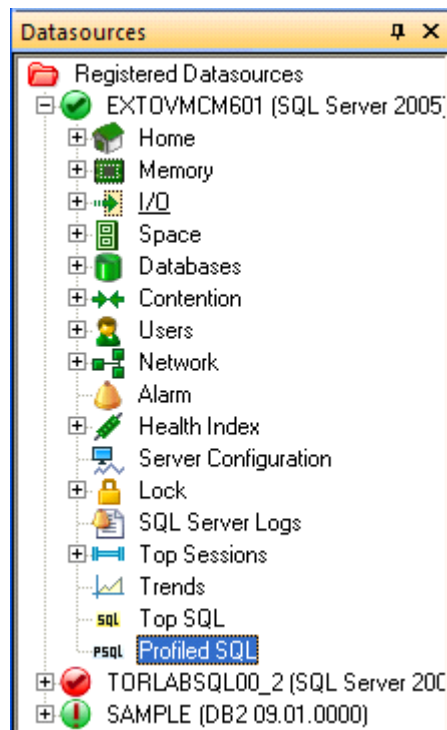
- When you select the SQL Capture option and click OK, a message appears asking if you want to test the selected data sources.



- Click **Yes** and after a brief moment the message closes. You will not get a confirmation on the compatibility. You will only get a warning message if it is not compatible.
- When an alert occurs on your selected statistic, an e-mail is generated.

Gaining Access to the Captured Information

- In the Datasource tree, double-click [Profiled SQL](#).



- In the SQL Profiling Session that opens for the selected datasource, select an Alert Name from the list box and click **Find**.

NOTE: You can also use a session ID as a search criteria.

- Select the result that you want information on and click **View Details**.

TORLABSQL00_2 - SQL Profiling Session - Find Session

Search Criteria

Session ID

Alert Name **Databases Needing Backup** (Critical) (ACTIVE) ▼

Filter by datetime interval

Begin End

SessionID	Alert Name	Severity	Session Started	Session Ended	Status
▶ 60	Databases Needing Backup	Critical	4/8/2009 7:41:15 AM	4/8/2009 8:11:15 AM	
64	Databases Needing Backup	Critical	4/8/2009 8:22:58 AM	4/8/2009 8:52:58 AM	
67	Databases Needing Backup	Critical	4/8/2009 9:04:19 AM	4/8/2009 9:34:19 AM	

Enterprise View | EXTOVMCM601 - Profiled SQL | EXTOVMCM601 - DATABASES | TORLABSQL00_2 - Profiled SQL

- In the Details tab that appears, you can scroll through the session details that have been generated. Note the color-coding of the Activity column allows you to quickly identify the type of activity.

TORLABSQL00_2 - SQL Profiling Session - Triggered by Databases Needing Backup on 4/8/2009 7:41:15 AM Session ID: 60

<< Back to Search

■ CPU
 ■ Lock
 ■ Memory
 ■ Buffer
 ■ I/O
 ■ Other

Statements	Executions	Activity (%)	User Name	Application
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Buffer...	9	32.14		
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Bu...	1	3.57	sa	Executor Module
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Bu...	1	3.57	sa	Executor Module
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Bu...	1	3.57	sa	Executor Module
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Bu...	1	3.57	sa	Executor Module
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Bu...	1	3.57	TEST_KAB	Executor Module
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Bu...	1	3.57	sa	Executor Module
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Bu...	1	3.57	TEST_KAB	Executor Module
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Bu...	1	3.57	sa	Executor Module
CREATE PROCEDURE sp_perfctr_primary AS BEGIN DECLARE @Bu...	1	3.57	sa	Executor Module
SELECT USER_NAME(O.uid),OBJECT_NAME(C.id),C.name,CASE WHEN C.xuser...	4	14.29		

```

1 CREATE PROCEDURE sp_perfctr_primary
2 AS
3 BEGIN
4     DECLARE @BufferCacheHrRatio    decimal(7,2),
5     @AdhocSQLHrRatio              decimal(7,2),
6     @ProcedurePlanHrRatio         decimal(7,2),
7     @LogCacheHrRatio              decimal(7,2),
8     @PreparedSQLPlanHrRatio       decimal(7,2),
9     @ExtendedMemoryCacheHrRatio   decimal(7,2),
10    @IOErrorRate                   decimal(7,2),
11    @PageSize                       int,
12    @BlockedUsers                   int,
13    @TotalConnections              int,
14    @ActiveConnections              int,
15    @InactiveConnections            int,

```

Enterprise View | EXTVOVMCM601 - Profiled SQL | EXTVOVMCM601 - DATABASES | TORLABSQL00_2 - Profiled SQL

Enabling SQL Capture for Oracle/Sybase/DB2/Microsoft SQL Server

- [Enable SQL Capture for Sybase ASE](#)
- [Enable SQL Capture for Oracle](#)
- [Enable SQL Capture for SQL Server](#)
- [Enable SQL Capture for DB/2](#)

Enable SQL Capture for Sybase ASE

Sybase ASE must be enabled if you want to use the capture function in Performance Center. Take the following steps to ensure that the SQL Capture feature is activated for Sybase ASE:

1 Enable the following System Configuration settings:

- Enabling Monitoring
- Wait Event Timing
- Max SQL Text Monitored
- SQL Batch Capture

Examples:

```
sp_configure "enable monitoring", 1
sp_configure "wait event timing", 1
sp_configure "SQL batch capture", 1
sp_configure "max SQL text monitored", 4096* *
```

NOTE: This is the maximum length for SQL text stored by Sybase, but we suggest 4096 as a good starting point.

2 Grant the user *mon_role*.

NOTE: If a user does not have *mon_role* enabled, the user will not be able to access Adaptive Server's monitoring tables.

3 Install the Sybase MDA tables (Sybase official documentation has directions - these should only be installed by a DBA).

- If the *monProcess* table is missing, the user will not be able to see currently connected sessions.
- If the *sysprocesses* table is missing, the user will not be able to see information about Adaptive Server processes.
- If the *monWaitEventInfo* table is missing, the user will not be able to see information about wait events.
- If the *monProcessSQLText* table is missing, the user will not be able to see currently executing SQL statements.

Example:

```
isql -U sa -P yourpassword -S YOURSERVER -i
$SYBASE/$SYBASE_ASE/scripts/installmontables
```

NOTE: Sybase 12.0 does not support SQL Capture.

Enable SQL Capture for Oracle

Perform the following tasks to ensure that Oracle is compatible with Performance Center:

- If you are setting up Oracle 10 or later, ensure you are logged in as **system or sys**, or the `SELECT_CATALOG_ROLE` has been granted to the connecting user.
- If you are setting up an earlier version of Oracle, ensure you are logged in as **system or* sys**

Enable SQL Capture for SQL Server

Perform the following tasks to ensure that SQL Server is compatible with Performance Center:

- If you are setting up SQL Server 8 or 9, ensure the current user is a member of sysadmin.
- If you are setting up later versions of SQL Server, the current user must meet one of the following requirements:
 - Be a member of sysadmin, or have the VIEW SERVER STATE permission enabled.
 - Be a member of sysadmin, or have the SELECT permission enabled.

Enable SQL Capture for DB/2

Perform the following tasks to ensure that DB/2 is compatible with Performance Center:

- Ensure that the following monitor flags are switched on:
 - BUFFERPOOL
 - LOCK
 - SORT
 - STATEMENT
 - TABLE
 - UOW
 - TIMESTAMP

Datasource Alerts

Datasource alerts are default or user-defined values that mark performance boundaries. When Performance Center detects threshold violations, it can do the following:

- Make an entry in the Alarm log
- Issue notification(s)
- Initiate an automated response or action

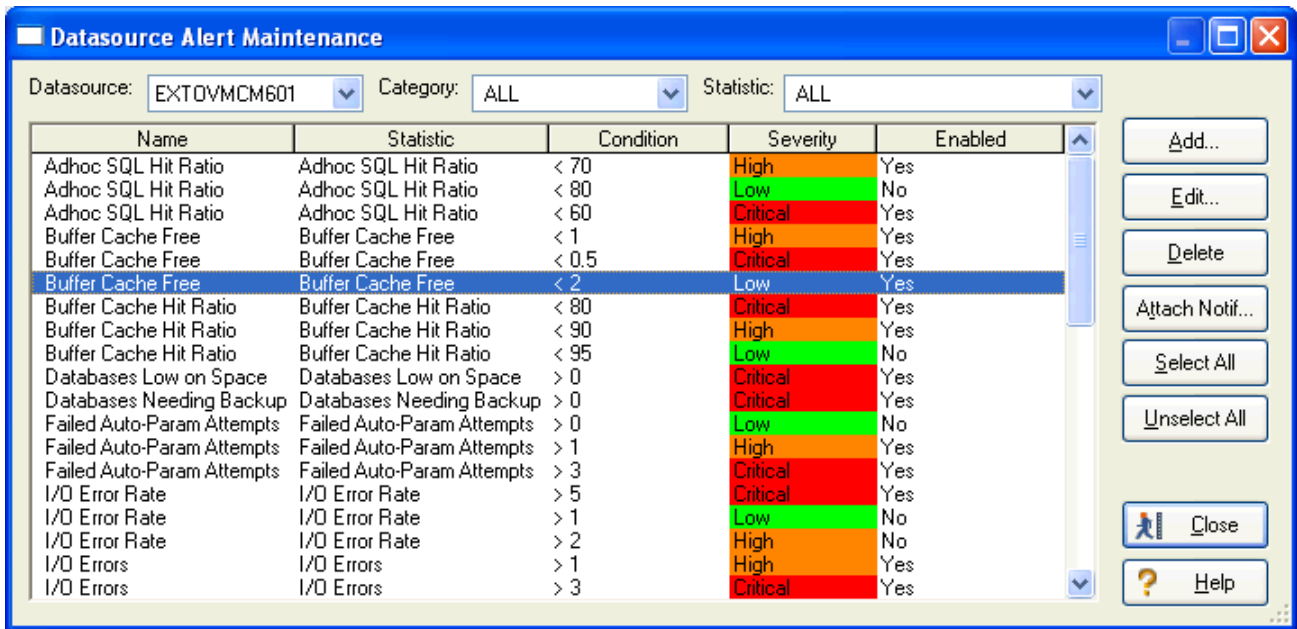
NOTE: These actions depend on what you define for the violated threshold.

Performance Center provides a Datasource Alert Maintenance dialog that includes three lists, which you can use to select the datasource, category of statistic, and statistic to view.

The table below describes the options and functionality available in the Datasource Alert Maintenance dialog:

Option	Description
Datasource	Select the target datasource you want to view from the available list.
Category	Select the target category you want to view from the available list.
Statistic	Select the target statistic you want to view from the available list.
Name	Displays a list of threshold names.
Statistic	Displays the statistic for each named threshold.
Condition	Displays the condition for each named threshold.
Severity	Displays the severity for each named threshold.

Option	Description
Enabled	Indicates whether each named threshold is enabled.



The Datasource Alert Maintenance dialog offers the following functionality for thresholds:

- [Add](#)
- [Edit](#)
- [Delete](#)
- [Attach Notification](#)

For more information, see:

[Adding an Alert](#)

[Editing an Alert](#)

[Deleting a Datasource Alert](#)

[Attaching an Alert to a Notification](#)

[Datasource Alerts Maintenance](#)

[Opening the Datasource Alerts Maintenance dialog](#)

Opening the Datasource Alerts Maintenance dialog

- Click **Datasource > Alerts Maintenance**.

Adding or Editing an Alert

Performance Center lets you add and edit [thresholds](#) for an Performance Center Server. To add or edit alerts, you must log on to the target Performance Center Server as an [Administrator](#).

You add or edit an alert using the Alert Properties dialog. The table below describes the options and functionality of the Alert Properties dialog:

Option	Description
Alert/Datasources	Displays the target alert or datasource.
Name	Lets you type or change the target alert name.
Description	OPTIONAL Lets you type or change a description for the target datasource.
Datasource Type	Choose the target DBMS type if you are adding a threshold. Displays the target DBMS type if you are editing a threshold.
Statistic	Choose the target statistic where you want to add threshold. Displays the target statistic if you are editing a threshold.
Generate Alarm if:	Choose or change the target alarm condition to: greater than, greater than or equal to, is equal to, is not equal to, is not equal to, less than, and less than or equal to.
Value amount	Type or change the numerical value for the threshold.
Severity:	Select the target severity level.
Enabled	Choose to enable the threshold.
Record Clear Event	Choose to make a record of when the threshold event clears.
SQL Capture	Select this option if you want to capture SQL text for the specified alert.
Notification	Attach a notification to the alert.
Action	Attach an action to the alert.
Blackout	Attach a blackout schedule to the alert.

For more information, see [Completing the Alert Properties dialog](#).

Related Topics

[Adding or Editing a Notification](#)

[Adding or Editing an Action](#)

Completing the Alerts Properties dialog

- 1 Click **Datasource > Alerts Maintenance**.
- 2 In **Datasource Alert Maintenance**, click **Add** or **Edit**. The Alert Properties dialog appears.
- 3 In the **Name** box, type the name for the new alert and the alerts are listed in the Alert/Datasources box.
- 4 **OPTIONAL** In the **Description** box, type the description for the alert.
- 5 In the **Condition** box, click the **Datasource Type** list, and then click the target DBMS.
- 6 Click the **Statistic** list, and then click the statistic on which to place the alert.
- 7 Click the **Generate Alarm if** list, and then click the condition for the threshold violation.
- 8 In the **Value** box, type the numerical value for the threshold violation.

- 9 Click the **Severity** list, and then click the target severity level/color.
- 10 To record when the threshold violation clears, select the **Record Clear Event** option.
- 11 To enable the threshold on completion of these steps, select the **Enabled** option.
- 12 To enable the SQL Capture, select the [SQL Capture](#) option.
- 13 To attach the threshold to a notification, click the **Notification** tab, and then:
Use the arrows to move the target notifications from the **Available Notifications** table to the **Selected Notifications** list.
OR
Click **Add**.
The Notification Properties dialog appears.
NOTE: For information on using the Notification Properties dialog, see [Adding or Editing a Notification](#).
- 14 To attach the threshold to an [action](#), click the **Action** tab, and then:
Use the arrows to move the target actions from the **Available Actions** table to the **Selected Actions** list.
OR
Click **Add** to open the Action Properties dialog.
NOTE: For information on using the Action Properties dialog, see [Adding or Editing an Action](#).
- 15 In the **Alert/Datasources** box, select the new Alert.
- 16 Click **Apply** and the Select Datasource dialog appears.
- 17 Select the target datasource.
- 18 Click **OK**. You are returned to the Alert Properties dialog and the datasource is added to the Alert list.
- 19 Click **OK**.

Deleting a Datasource Alert

You can delete [datasource alerts](#) from a Performance Center Server. To do this, you must log on to the target Performance Center Server as an [Administrator](#).

Deleting a Datasource Alerts

- 1 Click **Datasource > Alerts Maintenance**.
- 2 In the **Datasource Alert Maintenance** dialog, select the target alert.
- 3 Click **Delete** and a confirmation message appears.
- 4 In the message box, click **Yes**. The confirmation closes and the datasource alert is deleted.

[Adding a Recorded Statistic Template](#)

[Editing a Recorded Statistic Template](#)

[Deleting a Recorded Statistic Template](#)

[Cloning a Recorded Statistic Template](#)

[Applying a Recorded Statistic Template](#)

Alert Templates

Alert Templates are logical containers that define and deploy alerts to monitored datasources. The Alert Template Maintenance dialog is where you manage all Alert Templates for the all monitored datasources. The Alert Template Maintenance dialog offers the following functionality:

- [Add](#)
- [Edit](#)
- [Delete](#)
- [Clone](#)
- [Apply](#)
- [Attach Notification](#)

The table below describes the options and functionality available in the Alert Template Maintenance dialog:

Option	Description
View	Displays Alert Templates as the view.
Platform	Choose the target platform from the available list.
Name	View the name of each Alert Template
Description	See the description, if available, of each Alert Template.
System Template	Indicates if the template is a System Template.
Default Template	Indicates if the template is the Default Template.
Add	Opens the Alert Template Properties dialog where you can create new Alert Templates.
Edit	Opens the Alert Template Properties dialog where you can edit the selected Alert Template.
Delete	Removes the highlighted Alert Template from the Performance Center Server.
Clone	Opens the Clone Template dialog where you can create a new template based on an existing Alert Template.
Apply	Opens the Select Datasources dialog where you can select the datasource(s) to which to apply the highlighted Alert Template.
Attach Notif	Lets you attach one or more notifications to multiple thresholds.

Related Topics

[Alert Template Maintenance](#)[Recorded Statistic Templates](#)

For more information, see:

[Using the Alert Template Maintenance dialog](#)[Adding a Alert Template](#)[Editing a Alert Template](#)[Deleting a Alert Template](#)

[Cloning a Alert Template](#)

[Applying a Alert Template](#)

[Attaching a Alert Template to a Notification](#)

Recorded Statistic Templates

Recorded Statistic Templates are logical containers that provide a history of statistics of monitored datasources. Performance Center provides two dialogs to help you work with Recorded Statistics. The table below describes the dialogs:

dialog	Description
Recorded Statistic Template Maintenance	You can manage all recorded statistic templates for the all monitored datasources.
Datasource Recorded Statistics	You can view and select statistics that you want written to the Repository.

For more information, see:

[Recorded Statistic Template Maintenance dialog](#)

[Datasource Recorded Statistics dialog](#)

Recorded Statistic Template Maintenance dialog

The Recorded Statistic Template Maintenance dialog offers the following functionality:

- [Add](#)
- [Edit](#)
- [Delete](#)
- [Clone](#)
- [Apply](#)

The table below describes the options and functionality available in the Recorded Statistic Template Maintenance dialog:

Option	Description
Platform	Choose the target platform.
Name	See the name of each Alert Template.
System Template	Indicates if the template is a System Template.
Default Template	Indicates if the template is the Default Template.
Add	Opens the Recorded Statistic Template Properties dialog where you can create new recorded statistic templates.
Edit	Opens the Recorded Statistic Template Properties dialog where you can edit the selected recorded statistic template.
Delete	Removes the highlighted recorded statistic template from the Performance Center Server.

Option	Description
Clone	Opens the Clone Template dialog where you can create a new template based on an existing recorded statistic template.
Apply	Opens the Select Datasources dialog where you can select the datasource(s) to which to apply the highlighted recorded statistic template.

Related Topics

[Recorded Statistic Maintenance](#)

[Statistic Maintenance](#)

[Alert Templates](#)

For more information, see:

[Using the Recorded Statistic Template Maintenance dialog](#)

Using the Alert Template Maintenance Recorded Statistic Template dialog

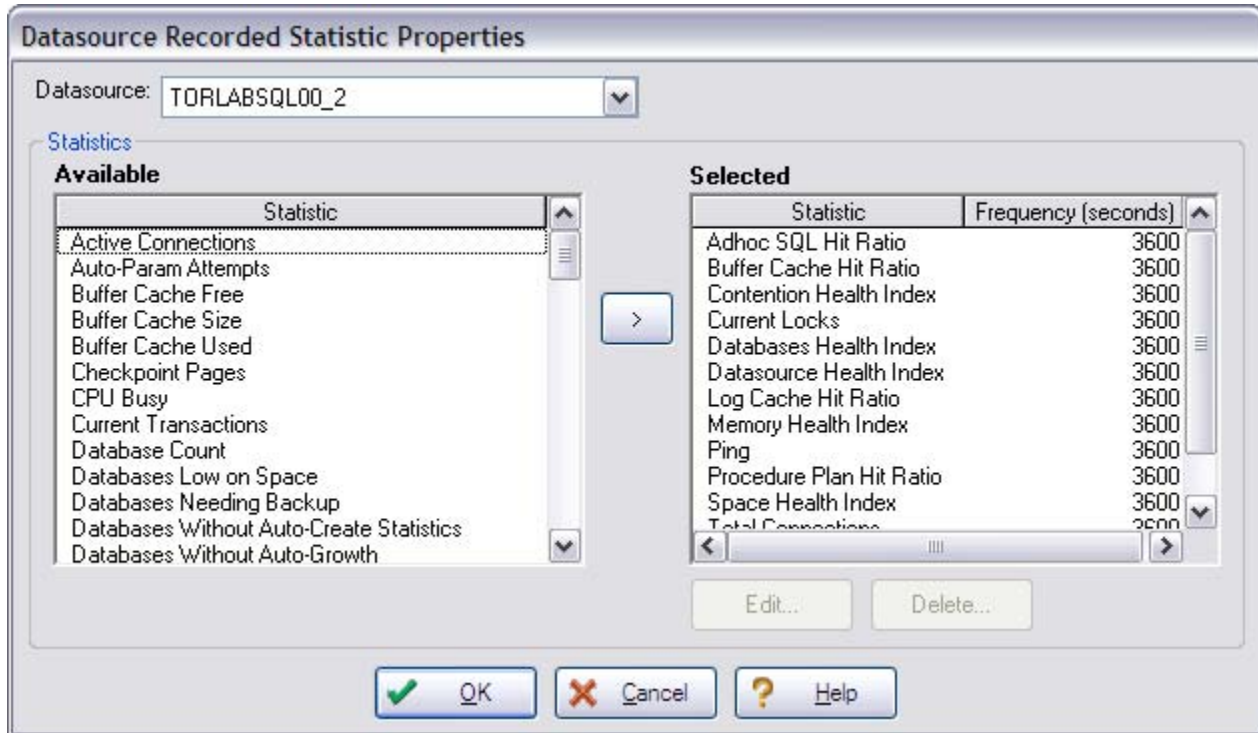
- 1 Click **Maintenance > Alert Templates** or **Recorded Statistic Templates**.
- 2 Select the target alert/statistics, and then click the target buttons.

Datasource Recorded Statistics dialog

You can view and select statistics that you want Performance Center to write to the Repository.

The table below describes the options and functionality available in the Datasource Recorded Statistic dialog:

Option	Description
Datasource	Choose the target datasource.
Available Statistics	See the available statistics on the target datasource.
arrow	Move statistics between Selected and Available.
Selected Statistic Name	See the statistics you want to write to the Performance Center Repository.
Selected Statistic Frequency	See the frequency (in seconds) that Performance Center will write the selected statistics to the Performance Center Repository.



Related Topics

[Recorded Statistic Maintenance](#)

[Statistic Maintenance](#)

[Alert Templates](#)

For more information, see:

[Using the Datasource Recorded Statistics dialog](#)

Using the Datasource Recorded Statistics dialog

- 1 Click **Datasource** > **Datasource Recorded Statistics**.
- 2 Select the target statistics, and then click the target buttons.

Adding or Editing an Alert or Recorded Statistic Template

Performance Center lets you add and edit [Alert Templates](#) and [recorded statistic templates](#) on an Performance Center Server. To add or edit templates, you must log on to the target Performance Center Server as an [Administrator](#).

You add or edit a Alert Template using the [Alert Template Properties](#) dialog; recorded statistic templates using the [Recorded Statistic Template Properties](#) dialog.

For more information, see:

[Alert Template Properties dialog](#)

Select Schedules

[Recorded Statistic Template Properties dialog](#)

Alert Template Properties dialog

The table below describes the options and functionality of the Alert Template Properties dialog:

Option	Description
Name	Type or change a template's name.
Description	OPTIONAL: Type or change a template's description.
Platform	Choose the target DBMS if you are adding a template. Displays the threshold's DBMS if you are editing a template.
System Template	Make the template a system-defined template.
Default Template	Choose to make the template a default template.
Available Alerts	View a list of available templates and add them to a new template or to the template you are editing. The list displays each template's name, alert, severity, and condition.
Selected Alerts	View the list of alerts in the template you are creating or editing.
arrows	Move alerts between the Available Alerts and Selected Alerts boxes.
Add	Opens the Alert Properties dialog where you can add Alert Templates.
Edit	Opens the Alert Properties dialog where you can edit existing Alert Templates.
Delete	Removes the highlighted Alert Templates from the Performance Center Server.

Related Topics

[Adding Alerts](#)

[Adding or Editing Alerts](#)

For more information, see [Completing the Alert Template Properties dialog](#).

Completing the Alert Template Properties dialog

- 1 Click **Maintenance > Alert Templates**.
- 2 In **Alert Template Maintenance**, click **Add** or **Edit**. The Alert Template Properties dialog appears.
- 3 In the **Name** box, type the name for the Alert Template.
- 4 **OPTIONAL** In the **Description** box, type the description for the Alert Template.
- 5 Click the **Datasource Platform** list, and then click the target DBMS.
- 6 To make the Alert Template a system-defined template, select the **System Template** option.
- 7 To make the Alert Template a default template, select the **Default Template** option.
- 8 Click the arrows to move target thresholds from the **Available Alerts** list to the **Selected Alerts** list.
- 9 To add thresholds, click **Add**. The [Alerts Properties](#) dialog appears.
- 10 To edit thresholds, select the target threshold on either list, and then click **Edit**. The [Alert Properties](#) dialog appears.

- 11 To delete thresholds, select the target threshold(s) on either list, and then click **Delete**. The confirmation message opens.
- 12 Click **Yes** and the target alerts are deleted.
- 13 When you finish adding or editing Alert Templates, click **OK**.

Recorded Statistic Template Properties dialog

The table below describes the options and functionality of the Alert Template Properties dialog:

Option	Description
Name	Type or change a template's name.
Description	OPTIONAL Type or change a template's description.
Platform	Choose the target DBMS if you are adding a template. Displays the threshold's DBMS if you are editing a template.
System Template	Make the template a system-defined template.
Default Template	Choose to make the template a default template.
Available Statistics	View a list of available statistics and add them to a new template or to the template you are editing.
Selected Statistics	View a list of available statistics and add them to a new template or to the template you are editing.
Frequency	Displays the time (in seconds) that the target statistic is written to the Performance Center Server.
arrows	Move statistics between Available Statistics and Selected Statistics boxes.
Edit	Opens the Select Schedule dialog where you can edit the recording schedule for the statistic highlighted in the Selected Statistics box.
Edit All	Opens the Select Schedule dialog where you can globally edit the recording schedule for all of the statistics in the Selected Statistics box.
Delete	Removes the highlighted recorded statistic from the recorded statistic template.

Related Topics

[Statistic Properties dialog](#)

[Schedules](#)

For more information, see:

[Completing the Recorded Statistic Template Properties dialog.](#)

[Select Schedule dialog](#)

Completing the Recorded Statistic Template Properties dialog

- 1 Click **Maintenance > Recorded Statistic Templates**.
- 2 In the **Recorded Statistic Template Maintenance** dialog, click **Add** or **Edit**. The Recorded Statistic Template Properties dialog appears.
- 3 In the **Name** box, type the name for the recorded statistic template.
- 4 **OPTIONAL** In the **Description** box, type the description for the recorded statistic template.

- 5 Click the **Datasource Platform** list, and then click the target DBMS.
- 6 To make the recorded statistic template a system-defined template, select the **System Template** option.
- 7 To make the recorded statistic template a default template, select the **Default Template** option.
- 8 Click the arrows to move target statistics from the **Available Statistics** list to the **Selected Statistics** list.
- 9 To edit the recording interval for a statistic, select the target statistic, and then click **Edit**. The [Select Schedule](#) dialog appears.
- 10 To edit the recording interval for all statistics, click **Edit All**. The [Select Schedule](#) dialog appears.
- 11 To delete statistics from the template, select the target statistic(s), and then click **Delete**. A confirmation message appears.
- 12 Click **Yes** and the target statistics are deleted from the template.
- 13 When you finish adding or editing templates, click **OK**.

Select Schedule dialog

The table below describes the options and functionality on the Select Schedule dialog:

Option	Description
Name	The name of the schedule.
Description	The brief description of the schedule.
System	Indicates if the schedule is a system schedule.
Add	Opens the Schedule Properties dialog where you can add a new schedule.
Edit	Opens the Schedule Properties dialog where you can edit an existing schedule.
Delete	Removes the highlighted recorded schedule from the Performance Center Repository.
Attach	Opens the Recorded Statistic Template Properties dialog where you can attach a schedule to a recorded statistic template.

For more information, see [Opening the Select Schedule dialog](#).

Opening the Select Schedule dialog

- 1 Click **Maintenance > Recorded Statistic Templates**.
- 2 In the **Recorded Statistic Template Maintenance** dialog, click **Add** or **Edit**. The Recorded Statistic Template Properties dialog appears.
- 3 In the **Available** box, select the target statistic, and then click the arrow. The statistic is moved to the Selected box.
- 4 In the **Selected** box, select the target statistic, and then click **Edit** or **Edit All**. The Select Schedule dialog appears.

Deleting a Alert Template or Recorded Statistic Template

You can delete [Alert Templates](#) and [recorded statistic templates](#) from an Performance Center Server. To delete templates, you must log on to the target Performance Center Server as an [Administrator](#).

Deleting a Alert Template or Recorded Statistic Template

- 1 Click **Maintenance > Alert Templates** or **Recorded Statistic Templates**.
- 2 In the **Alert Template Maintenance** or **Recorded Statistic Template Maintenance** dialog, select the target template(s).
- 3 Click **Delete** and a confirmation message appears.
- 4 In the message box, click **Yes** and the confirmation closes and the template(s) is deleted.

Cloning an Alert Template or Recorded Statistic Template

You can [clone](#) Alert Templates and recorded statistic templates for an Performance Center Server. To clone templates, you must log on to the target Performance Center Server as an [Administrator](#).

You clone templates using the Clone Template dialog. The table below describes the options and functionality of the Clone Template dialog:

Option	Functionality
Template Name	Lets you type the name for the threshold you want to create.
Copy properties from	Lets you choose the target threshold you want to clone.



For more information, see:

[Opening the Cloning Alert Template or Recorded Statistic Template dialog.](#)

Opening the Clone Alert Template or Recorded Statistic Template dialog

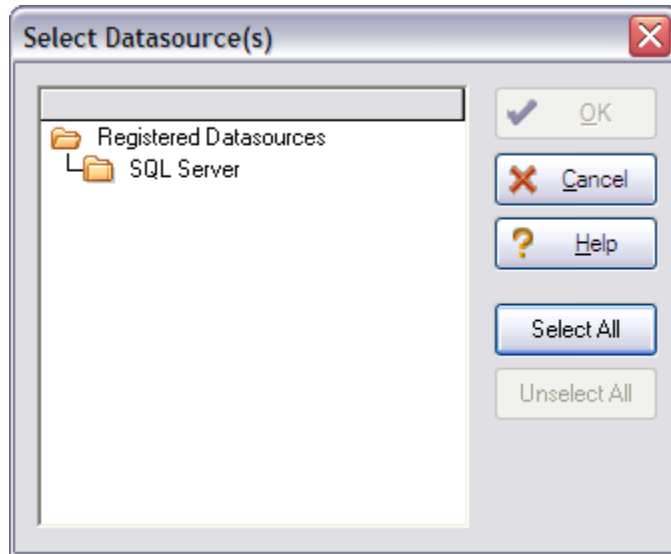
- 1 Click **Maintenance > Alert Templates** or **Recorded Statistic Templates**.
- 2 In the **Alert Template Maintenance** or **Recorded Statistic Template Maintenance** dialog, click **Clone**. The Clone Template dialog appears.

Applying an Alert, Health Index Template, or Recorded Statistic Template

You can apply any alert, Health Index template or recorded statistic template to an Performance Center Server. To apply, you must log on to the target Performance Center Server as an [Administrator](#).

Applying an Alert, Health Index Template, or Recorded Statistic Template

- 1 Click **Maintenance > Alert Templates, Health Index Templates** or **Recorded Statistic Templates**.
- 2 In the **Alert Template, Health Index Template Maintenance, or Recorded Statistic Template dialog**, select the target templates.
- 3 Click **Apply** and the Select Datasources dialog appears.



- 4 Select the target datasource(s).
- 5 Click **OK** and the template is applied to the target datasource(s).

Reports

Reports are detailed historical accounts of performance information. The Performance Center reporting feature lets you determine database downtime, the best and worst performing databases in your organization, historical traffic volumes, times that are the busiest for a database, and much more.

Performance Center uses the Microsoft Task Scheduler component of IE for scheduling some reports. You must have the start the Scheduler service to schedule Performance Center reports.

The Report Maintenance dialog lets you automatically generate detailed historical reports on all monitored datasources and save them to any location for future reference.

The table below describes the options and functionality of the Report Maintenance dialog:

Option	Description
Name	Displays the name of each report.
Description	Displays a brief description of each report.
Scheduled	Indicates if the report is scheduled .
Add	Opens the Select Historical Report dialog where you can create new reports.
Edit	Opens the Custom Report dialog where you can edit the selected report.
Delete	Removes the highlighted report from the Performance Center Server.

Related Topics

[Report Menu](#)

[Schedules](#)

For information, see:

[Using the Report Maintenance dialog](#)

[Select Historical Report Wizard](#)

[Custom Report dialog](#)

[Logging into the Oracle Statspack](#)

[Report Maintenance](#)

Using the Report Maintenance dialog

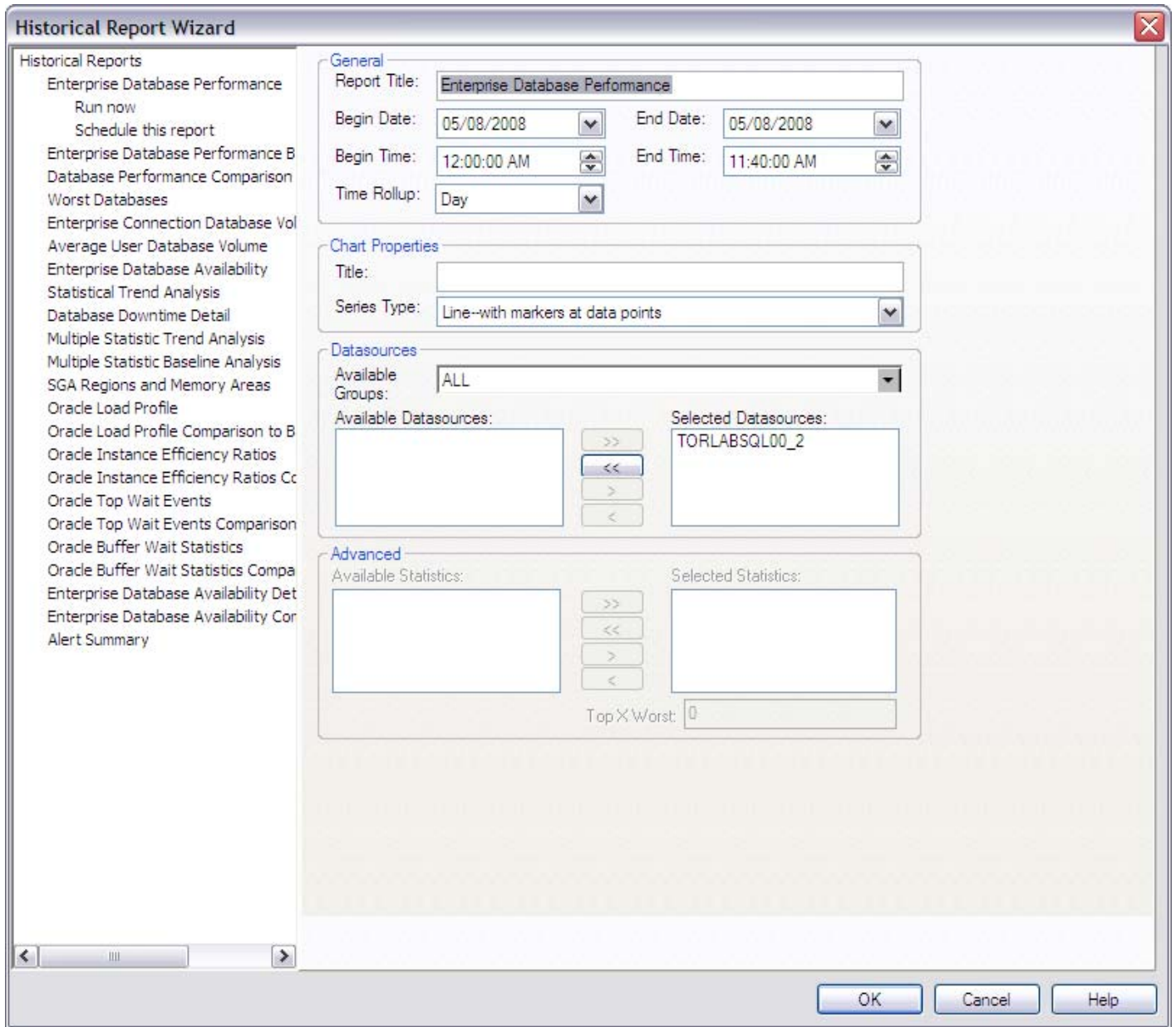
- 1 Click **Maintenance > Reports**.
- 2 Select the target reports and click the target buttons.

For information, see:

[Reports](#)

Select Historical Report Wizard

The **Historical Report Wizard** provides a means to view, generate, and schedule reports.



For more information, see:

[Completing the Select Historical Report dialog](#)

Completing the Historical Report Wizard

- 1 Click **Report > Historical**.
- 2 Select a report from the list provided, depending on the type of report you want to view or generate.
- 3 Modify the report parameters, as needed.
- 4 Click **OK** to run the report. When the report is finished generating, it appears in a separate tab in the Performance Center environment.

For more information, see:

[Reports](#)

Logging into the Oracle Statspack

You can quickly enter user/password information so that you can log into the Oracle Statspack to help you run reports.

NOTE: The Oracle Statspack menu item is available if an Oracle 8 or later datasource is highlighted in Explorer window.

For more information, see:

[Report Menu](#)

[Report Maintenance](#)

Data Extraction Jobs

You can export data so you can make statistical information readily available when you have time and energy to review the findings. The procedures available let you migrate repository or alarm data. Another feature is that you can export the data across platforms so that you can take information gathered on an Oracle datasource and move it to a table on a Sybase database. You can export data to an existing table or create a new table to contain the data. Also, you can schedule the export operation to occur once or to be performed on a recurring basis.

You can also use the exported data to create customized reports from the Web Client. The Reports function on the Web Client lets you generate reports remotely for data you have previously exported as part of any data extraction job. No special permissions are required. This gives you the opportunity to view reports wherever you have access to the Web Client. It is important to keep this in mind when you schedule a data extraction job. If you are planning to run reports from a remote client, keep in mind that you cannot change the particulars of an extraction job from a distance. Any report you run will be based on the most recently available data.

The [Data Extraction Jobs](#) dialog is the point of origin for all export operations in Performance Center. Using this dialog, you can Add, Edit, or Delete export tasks. For export tasks you have previously created and scheduled, there is also a [Run Now](#) option so you can export data at a one-off interval.

To open the Data Extraction Jobs functionality:

- 5 Click **Maintenance > Data Extraction Jobs** and the Data Extraction Jobs dialog appears.

Data Extraction Jobs dialog

The Data Extraction Jobs dialog is the locus of all export operations. It's here that you can you can **Add**, **Edit**, or **Delete** export tasks. When you open the Data Extraction Jobs dialog, you see a list of all the export jobs you have previously created and you can isolate one or more for tweaking. And, for export tasks you have previously created and scheduled, there is also a [Run Now](#) option so you can export data at a one-off interval.

To Add or Edit an export data task:

- 6 Click **Maintenance > Data Extraction Jobs** and the Data Extraction Jobs dialog appears.
- 7 Click **Add** or **Edit** as appropriate, and the [Data Extraction Job Properties](#) dialog opens.

Data Extraction Job Properties dialog

The Data Extraction Job Properties dialog is where you determine what kind of export/migration operation you want to perform and the parameters of the “to” and “from” database and datasource statistics. After a task has been created and added to your maintenance schedule, you can easily edit all aspects of the job using this same dialog.

The table below describes the fields you’ll find on the Data Extraction Job Properties dialog:

Field	Description
Job Name	The name of the data extraction job.
Job Name	This is where you name the export job. You should consider using the name of the table where you are migrating the data to avoid confusion (this is only a suggestion, however).
Task Type	You have the choice of two task types. You can either export repository data or alarm data.
Time Rollup	You can specify whether or not you want to see data that was collected at specified intervals--minute, hour, day, week, or month, for the number of days you specify in the field that follows. If you choose None, you’ll get the data without any rollups.
At execution time, gather data from the last x days	The default is 7 days. The greatest number of days for which you can collect data is one year, i.e., 365 days.
Location of Extraction Data	In this section of the dialog you are specifying where you want the data to be exported to.
Datasource Type	Datasource platform is what’s meant, and you can choose among Oracle, SQL, and Sybase from the list.
Datasource Name	The list you find here is populated with datasources that are registered on the platform you selected. You can also type the name of a registered datasource.
User Name	You must be able to connect to the datasource with this User Name. If you want to add a new user, you can only do so if you have administrator privileges. Click New User to add a user.
Password	Type the password associated with the specified user.
Database Name	Identify a database where you want the table of exported data to reside. NOTE: For the job to run and make the world a safer place for data, you must use the Test Connection button and verify that the test login has succeeded.
Table Name	This is where you name the table where you want to export the data. You can type the name of a table that already exists or you can type the name of a table you want to create.
Extracted Statistics	In this section you make selections about the source and data you want to export to the location you identified in the section above.
Platform Filter	Like Datasource Type above, you can choose among Oracle, Sybase, and SQL. You can export data across platforms, which is a pretty nifty feature.
Available and Selected Datasources:	You’ll find a list of all datasources that are registered in Performance Center for the platform filter you specified. You can select one or more of them, click an arrow, and move your choice(s) into the Selected Datasources field.
Available and Selected Statistics	Lets you pick one or more of the statistics available for monitoring on the specified platform. For example, an Oracle datasource doesn’t have all the same statistics as a SQL platform. This does not affect the export operation or the creation of the table where data is being exported.
Schedule	NOTE: For the export operation to succeed, you must create a schedule even if you only want to export the data once. When you click the Schedule button, as you must to be successful, the Schedule Properties dialog opens.

Adding a New User for a Reporting Repository

If you want a new user to have access to the datasource where you are sending the exported data, you can register a new user from the task interface if you have administrator privileges. The new user is then able to review the data you have migrated.

When you click **New User**, the User-Password dialog opens. This dialog merely ascertains that the person adding the new user has the appropriate administrator privileges to make such grants. The administrator can use OS authentication or use database authentication to receive validation.

After Performance Center has validated the administrator privileges, a platform-based wizard that lets you add new users opens. You may be familiar with this wizard from Embarcadero's DBArtisan application. If not, that information is repeated here:

- [Login Wizard for Microsoft SQL Server](#)
- [Create User Wizard for Oracle](#)
- [Login Wizard for Sybase ASE](#)

Login Wizard for Microsoft SQL Server

Panel 1

The table below describes the options and functionality on the first panel of the Login Wizard.

Option	Description
What is the name of the login?	Type the login name.
Specify whether this login account will be for a SQL Server or a Windows NT User or Group.	<p>If the this is a Microsoft SQL Server Login, click the SQL Server option button and complete the password and confirm boxes.</p> <p>Microsoft SQL Server 7 or later If the this is a NT User or Group Login, click the NT User or Group option button.</p> <p>Click the Domain list, and then click a domain.</p>

Panel 2

The table below describes the options and functionality on the second panel of the Login Wizard.

Option	Description
What should be the login's default database?	Select the default database. Avoid specifying the master database to prevent users from creating objects there inadvertently.
What is the login's default language?	Select the default language.

Panel 3

The table below describes the options and functionality on the second panel of the Login Wizard.

NOTE: The options on this panel differ by version of Microsoft SQL Server.

Option	Description
Select database to create users	Select the appropriate target user options or click check all. User Name - If you are adding a user name that is different from the Login ID, type the new name in the User Name box. Database Roles - Select the options that correspond to the target databases or click Select All.

Create User Wizard for Oracle

Panel 1

The table below describes the options and functionality on the first panel of the User Wizard.

Option	Description
What is the name of the user?	Type the name of the user. It can be up to 30 characters long.
How should the user be identified box?	Password - Indicates that Oracle should identify the user with the password you provide. In the Password box and in the Confirm box, type the password for the user. Externally - Indicates that Oracle should verify the database user name against an existing operating system user name. Globally - Indicates that Oracle permits access to the user by obtaining username and password information from the security domain central authority. ORACLE 8 ONLY: External Name - Type the external name of the database user.
What access do you want to grant to the user's account?	NOTE: Only available for Oracle 8. Locked Unlocked

Optional Panel 2

The table below describes the options and functionality on the second panel of the User Wizard.

Option	Description
What is the default tablespace for objects that the user creates?	Select the default tablespace.
What is the tablespace to be used for the user's temporary segments?	Select the tablespace.
What is the profile for the user?	Select the profile.

Optional Panel 3

The table below describes the options and functionality on the third panel of the User Wizard.

Option	Description
Set Quota Button	In the Tablespace box, click the tablespace you want, and then click the Set Quota option button.
Select All Button	Select all tablespaces.
Unselect All Button	If you have selected any tablespaces you can in the Tablespace box, click the Unselect All option button.

Optional Panel 4

The table below describes the options and functionality on the fourth panel of the User Wizard.

Option	Description
Do you wish to grant any roles to the user? If so, then select them from the list below.	Select the roles, if any, to grant to the user.
Do you wish to grant admin options to all selected roles?	Grant admin options to all roles.
Do you wish to enable all selected roles as default roles?	Mark the selected roles as the defaults.

:Login Wizard for Sybase ASE

Panel 1

The table below describes the options and functionality on the first panel of the Login Wizard.

Option	Description
What is the ID for the Login?	Type the login name. Can be up to 30 characters long.
What is the password for the new login?	Can be up to 30 characters long, but must be at least six characters.

Panel 2

The table below describes the options and functionality on the second panel of the Login Wizard.

NOTE: You should avoid specifying the master database to prevent users from creating objects there inadvertently.

Option	Description
What is the full name of the login box?	OPTIONAL: Type the full name for the login (for example, John Smith).
What should be the login's default database?	Database - Select the default database.
What is the login's default language?	Language - Select the default language. If you do not select one, the server's default language is used.

Panel 3

The table below describes the options and functionality on the third panel of the Login Wizard.

Option	Description
Select any roles that you wish to grant to the login from the list below	OPTIONAL: Select the options that correspond to the roles you want to grant, or click the Select All button.
Do you want to lock the new login after creating it?	Lock the new login so that nobody else can use it.

Panel 4

The table below describes the options and functionality on the fourth panel of the Login Wizard.

Option	Description
Select databases to create users	Select the options that correspond to the databases or click the Check All button. User Type - Select the user type you want to add. If you select Alias in the Alias box, click the User Name list, and then click the user name. If you select dbo and you want to transfer aliases and permissions, click the Transfer Aliases and Permissions option. If you select User, in the User box, click the User Name list, and then click the user name, then click the Group list, and then click the user group.

Optional Panel 5

The table below describes the options and functionality on the Login Wizard.

Option	Description
Enter the password expiration, if any	Enter a value between zero and 32767 days.
Enter the minimum password length, if any	Enter a value between zero and 30.
Enter the maximum number of login attempts, if any	Enter a value between zero and 32767.

Deleting an Export Job

If an export job has outlived its usefulness, and it's not worth editing, it can easily be deleted. All you need to do is open the [Scheduled Report Jobs dialog](#), and click **Delete**.

Run Now...

To be functional, all export jobs are scheduled. But, if you want to run a job at the drop of hat, you can run the job at any time. To run a job at a non-scheduled interval, all you need to do is open the [Scheduled Report Jobs dialog](#), and click **Run Now**.

Archive

The Archive view lists your archived log files for all Oracle monitored databases. You can set your viewing parameters based on the following criteria:

- Today
- Last Two Days
- Last Week
- Last Two Weeks
- Last Month
- Last Two Months
- Last Three Months
- Last Six Months
- Full Log

NOTE: The Archive view is only available for Oracle datasources.

Using the Archive View

- 1 Click **View > Datasource > Archive**.
- 2 To filter the Archive view, click the **Show Archive Files for:** list, and then click the target filter.

TIP: To sort the list of logs, double-click the appropriate column heading.

Alarms

You can get alarms for all monitored datasources, so that you can view them. You can acknowledge any alarm that the Performance Center Server has sent a notification.

You use the Alarms Criteria dialog to set alarm search parameters.

For more information, see:

[Alarms Criteria dialog](#)

[Completing the Alarms Criteria dialog](#)

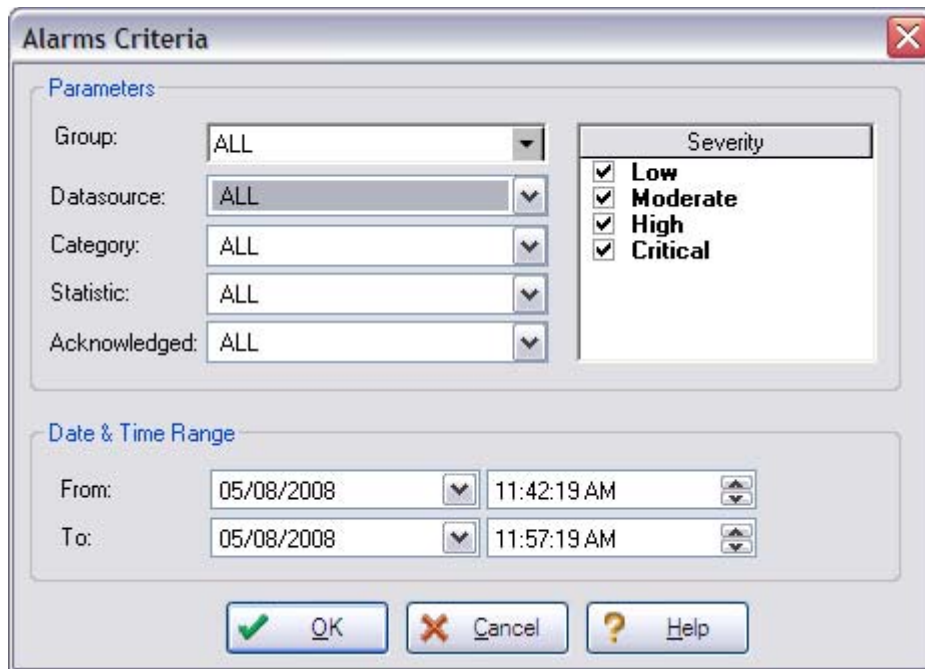
[Acknowledging Alarms](#)

Alarms Criteria dialog

The table below describes the options and functionality of the Alarms Criteria dialog:

Option	Description
Group	Select the target datasource group where you want to search for alarms.
Severity	Specify the target alarm levels you want to search for: Low, Moderate, High, or Critical.
Datasource	Select the target datasource where you want to search for alarms.

Option	Description
Category	Select the target categories where you want to search for alarms: All categories, Contention, Databases, I/O, Instance, Lock, Memory, Network, Operating System, Space, Top Sessions, and Users.
Statistic	Select the target statistics where you want to search for alarms.
Acknowledged	Select the target types of acknowledgements for the alarm search: Yes, No, or All.
From	Select the start date and time when you want to begin the search for alarms.
To	Select the end date and time when you want the search for alarms to complete.



For more information, see:

[Completing the Alarms Criteria dialog](#)

[Acknowledging Alarms](#)

Completing the Alarms Criteria dialog

- 1 On any view, click the **Alarm** icon located in the upper right-hand corner of the screen.
- 2 Click **Alarm > Get Alarms** and the Alarms Criteria dialog appears.
- 3 Click the **Group** list, and then click the target datasource group for the alarm search.
- 4 Click the **Datasource** list, and then click the target datasource for the alarm search.

TIP: To search all target parameters, click All.
- 5 Click the **Category** list, and then click the target categories for the alarm search.
- 6 Click the **Statistic** list, and then click the target statistics for the alarm search.

- 7 Click the **Acknowledged** list, and then click the target types of acknowledgements for the alarm search.
- 8 In the **Severity** box, select the options to specify the severity levels for the alarm search.
- 9 In the **Date & Time Range** box, click the **From** lists to specify the start range for the alarm search, and **To** to specify the end range for the alarm search.
- 10 Click **OK**. Performance Center searches for alarms meeting the criteria and then populates the Alarm table with the alarms.

TIP: To sort the list of alarms, double-click the appropriate column heading.

Acknowledging Alarms

Performance Center lets you acknowledge alarms by including a note in the repository. You use the [Alarm Notifications](#) and the [Alarm Acknowledgment](#) dialogs to do this.

For more information, see:

[Alarm Notifications dialog](#)

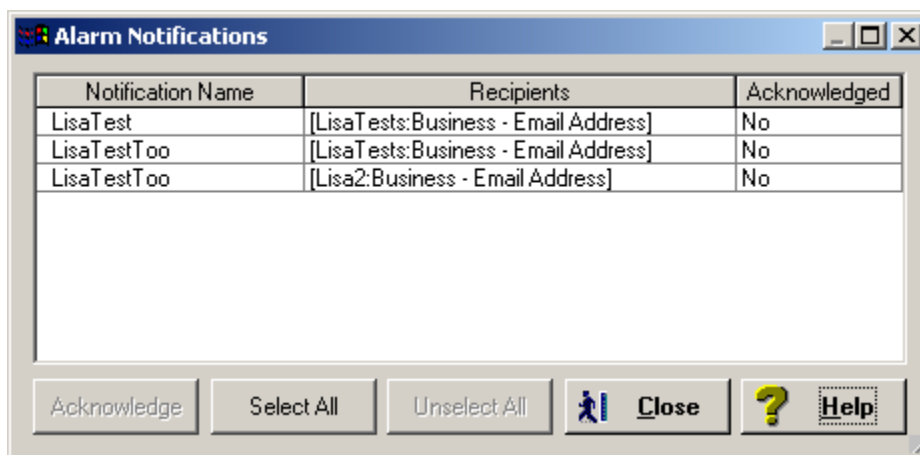
[Alarm Acknowledgements dialog](#)

[Adding Alarm Acknowledgements](#)

Alarm Notifications dialog

The table below describes the options and functionality of the Alarm Notifications dialog:

Option	Description
Notification Name	View the list of notifications. To add a notification, see Adding or Editing Notifications .
Recipients	View the recipients for each notification.
Acknowledged	See if each notification is acknowledged or not.



Related Topic

[Adding or Editing Notifications](#)

For more information, see:

[Alarm Acknowledgements dialog](#)

[Adding Alarm Acknowledgements](#)

Alarm Acknowledgements dialog

The table below describes the options and functionality of the Alarm Acknowledgements dialog:

Option	Description
Resolution:	You can specify the alarm resolution type for the repository. The resolution types include: Ignored, Natural, Auto-Connected, and Manually Corrected.
Note:	Type additional information to include in the alarm acknowledgment.

Adding Alarm Acknowledgements

- 1 [Open](#) the **Alarm** view.
- 2 In the grid, right-click the target alarm(s), and then click **Acknowledge** and the Alarm Notifications dialog appears.

NOTE: The Alarm Notifications dialog opens only when multiple alarm notifications are available. If only one notification is available, Performance Center skips to step 4.
- 3 In the **Alarm Notification** dialog, select the target alarm notification(s), and then click **Acknowledge**. The Alarm Acknowledgement dialog appears.
- 4 In the **Resolution** box, enter the target type of acknowledgement.
- 5 In the **Note** box, enter any additional information.
- 6 Click **OK**.

Alert Log/SQL Server Logs/Sybase Error Log/DB2 Diagnostic Log

The following views are available for viewing alert lists:

- Oracle Alert Log
- SQL Server Log
- Sybase Error Log
- DB2 Diagnostic Log

All views include filters, which enable you to customize your display for specific logs.

Oracle Alert Log

To view the Oracle alert log, the following prerequisite must be met:

The directory that contains the alert log must be specified in the instance parameter initialization file (INIT<SID>.ORA) using the UTL_FILE_DIR parameter.

For example:

(UNIX-Specific)

UTL_FILE_DIR =/server/orainventory/admin/linux1/bdump

(Windows Specific)

UTL_FILE_DIR =d:\oracle\admin\pmanest\bump

NOTE: After the UTL_FILE_DIR parameter has been specified, the Oracle datasource has to be restarted for the change to take effect.

Sybase Error Log

The Sybase Error Log functionality makes use of xp_cmdshell and grep to query the error log. When you monitor a Sybase datasource that resides on Windows, the Sybase Error Log Functionality is only supported if grep resides on Sybase datasource machine path.

NOTE: You may have to start the XP server on UNIX manually. You can either set the xp_cmdshell context to 0 by running the command sp_configure"xp_cmdshell_context",0 or an operating system user account must exist for the Sybase user name.

DB2 Diagnostic Log

The diagnostic log file contains text information logged by DB2. The information can be used to determine problems. If you suspect there is a problem on a specific machine, you can open the db2diag.log file on that machine. Information is appended to the end of the log, so the most recent items will be at the end of the file.

Alert View Elements

The table below describes the functions available on the Alert Log, SQL Server Log, and Sybase Error Log views:

Element	Action
Show List	Specify the display of the log. SYBASE ONLY Specify the date for the log.
For List	ORACLE & SQL SERVER ONLY Specify the target time period for messages in the chart.
Find	Opens the Find dialog .
Log Path	SYBASE ONLY Opens the Sybase Error Log Information dialog.
Refresh	ORACLE ONLY Opens the Alert Log Information dialog . SQL SERVER & SYBASE ONLY Refreshes the information on the SQL Server view.

For more information, see:

[Completing the Find dialog in the Alert Log/SQL Server Logs View](#)

[Completing the Alert Log Information dialog](#)

[Completing the Sybase Error Log Information dialog](#)

Opening the Alert Log/SQL Server Logs/Sybase Error Log

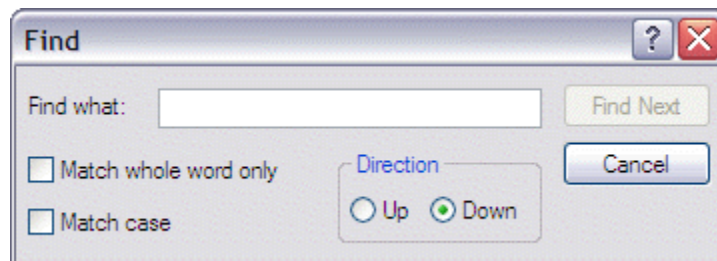
- 1 On the **View** menu, click **Alert Log/SQL Server Logs/Sybase Error Log/DB2 Diagnostic Log**.
- OR
- 2 On any view, click one of the DBMS-specific buttons in the upper right-hand corner of the view. For example, on **SQL Server** views, click **SQL Server Logs**.

TIP: To sort the list of alert logs, double-click the appropriate column heading.

Completing the Find dialog in the Alert Log/SQL Server Logs/Sybase Error Log View

Performance Center lets you specify a search string to help you find alerts.

- 1 On the **Alert Log/SQL Server Logs/Sybase Error Log** view, click **Find**.
Performance Center opens the Find dialog.



- 2 In the **Find what** box, type the search string.
- 3 To narrow the search, select any of the options or option buttons.
- 4 Click **Find Next** and a search is done for the target search string.

Completing the Alert Log Information dialog

Performance Center lets you enable the alert log from the Alert Log view.

- 1 On the **Alert Log** view, click **Refresh**.
Performance Center opens the Alert Log Information dialog.
- 2 In the **Log File Path** box, type the log file path (including the directory and file name) for the Oracle instance.
- 3 Click **OK**.
- 4 To enable the Alert Log, you must do the following:
 - Add the following entry to the Init.ora file for the target datasource: `UTIL_FILE_DIR = alert log directory name`
 - Restart the target datasource.

Completing the Sybase Error Log Information dialog

Performance Center lets you enable the Log Path from the Sybase Error Log view.

- 1 On the **Sybase Error Log** view, click **Log Path** and the Sybase Error Log Information dialog appears.
- 2 In the **Log File Path** box, type the log file path (including the directory and file name).
- 3 Click **OK**.

Health Index

The Health Index view is a historical display of all [Health Indexes](#) for the current datasource.

For more information, see [Health Index View](#).

Opening the Health Index View

- Click **View > Datasource > Health Index**.

Instance Parameters/Server Configuration/Configuration

The Instance Parameters/Server Configuration/Configuration view displays a list of all datasource configuration parameters for the current datasource.

Using the Instance Parameters/Server Configuration View

- **ORACLE ONLY:** On the **View** menu, click **Datasource > Instance Parameters**.
- **SQL SERVER & SYBASE ONLY:** On the **View** menu, click **Datasource > Server Configuration**.
- **DB2 ONLY:** On the **View** menu, click **Datasource > Configuration** and the Instance Parameters/Server Configuration view appears.

TIP: To sort the list of datasource parameters, double-click the appropriate column heading.

Locks

The Locks view displays all active lock information for the current datasource. It includes the following tabs:

- All Locks
- All User Locks (ORACLE ONLY)
- Blocking Locks

To view specific lock categories, click the appropriate tab.

For more information, see [Locks View](#).

Using the Locks View

- 1 On the **View** menu, click **Locks** and the Locks view appears.
- 2 To view specific lock categories, click the appropriate tab.

TIP: To sort the list of locks, double-click the appropriate column heading.

Object Extents Tab

The Object Extents tab of the Objects Detail view lets you view all object extents for all Oracle monitored databases. You can view the number of extents from the maximum:

NOTE: The Object Extents tab of the Objects Detail view is only available for Oracle datasources.

Using the Object Extents tab of the Objects Detail View

- 1 On any Oracle view, right-click the performance category toolbar, and then click **Object Extents**.
- 2 To view the number of extents from the maximum, in the **Enter # of Extents from Max:** box, type the target value.

TIP: To sort the list of extents, double-click the appropriate column heading.

Top Sessions

The Top Sessions view displays all sessions in the current datasource. The Top Sessions view displays each category on a tab, which you can sort by double-clicking the appropriate column. All sessions are ranked in I/O or memory priority, displaying the top session hogs first.

For more information, see [Top Sessions View](#).

Using the Top Sessions View

- 1 On the **View** menu, click **Top Sessions**.
- 2 To view specific session categories, click the appropriate tab.

NOTE: The available tabs depend on the target database platform.

TIP: To sort the list of sessions, double-click the appropriate column heading.

Session Detail

The Session Detail view displays details for all sessions in the current datasource. The Session Detail view displays each category on a tab, which you can sort by double-clicking the appropriate column. All sessions are ranked in I/O or memory priority, displaying the top session hogs first.

For more information, see [Session Detail View](#).

Opening the Session Detail View

- 1 Click **View > Top Sessions** and the Top Sessions view appears.
- 2 Right-click one of the rows in one of the tables, and then click **Session Details** to open the Session Detail view.
- 3 To view specific session details, click the appropriate tab.

NOTE: The available tabs depend on the target database platform.

TIP: To sort the list of sessions, double-click the appropriate column heading.

The table below describes the information available on the SQL tab of the Session Detail view:

Element	Description
SQL Statement box	Displays the SQL for the target session.
Other SQL box	ORACLE ONLY Displays the SQL for other current sessions.
Explain SQL button	ORACLE ONLY Runs an explain plan on the target SQL.
Query Plan button	SQL SERVER & SYBASE ONLY Runs a query plan on the target SQL.
Tune SQL button	If installed, starts the Embarcadero SQL Tuner.
Rows Returned	SQL SERVER ONLY Displays the number of rows returned by the target query.

For more information, see [Top Sessions View](#).

Top SQL

The Top SQL view displays all SQL running or executed in the current Oracle, SQL Server, DB2 or Sybase datasource. You can use the lists and boxes to reorder the display according to your specific needs. The table below describes the functions available on the tabs of the Top SQL view:

Button	Action	Tab
Show SQL	Specify the display of the top SQL.	Criteria
Reset	Reset the search parameters for a top SQL search.	Criteria
Find	Opens the Find dialog that lets you search for specific strings in the result set.	Results
Explain SQL	ORACLE ONLY Runs an EXPLAIN plan on the target SQL.	Results
Query SQL	SQL SERVER ONLY Runs a QUERY plan on the target SQL.	Results

For more information, see:

[Customizing the Top SQL Display](#)

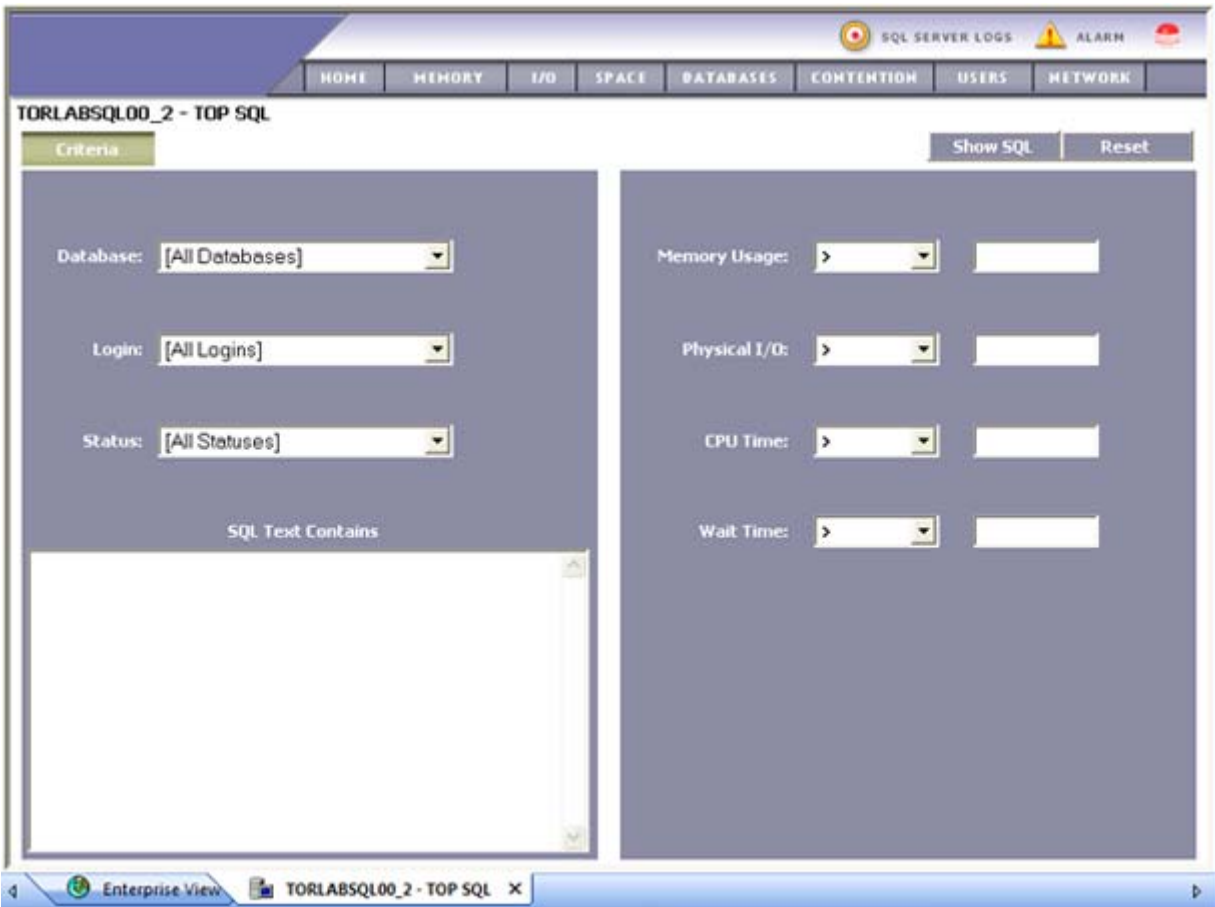
[Completing the Find dialog in the Top SQL View](#)

[Running an Explain Plan/Query Plan on SQL Statements](#)

[Top SQL View](#)

Opening the Top SQL View

- Click **View > Datasource > Top SQL** to open the Top SQL view.



Customizing the Top SQL Display

Performance Center lets you customize the display of the top SQL statements.

- 1 On the **Top SQL** view, click the lists, and then type the appropriate number(s) in the boxes.
- 2 Click **Show SQL** to open the Results tab and display the target SQL.
- 3 To sort the **Results** grid, double-click the appropriate column heading.
- 4 To specify another Top SQL view, click **Criteria** and the Criteria tab of the Top SQL view appears.
- 5 Click **Reset** and the search criteria on the Top SQL view resets to the defaults.

NOTE: To customize your Top SQL request, you must click Reset.

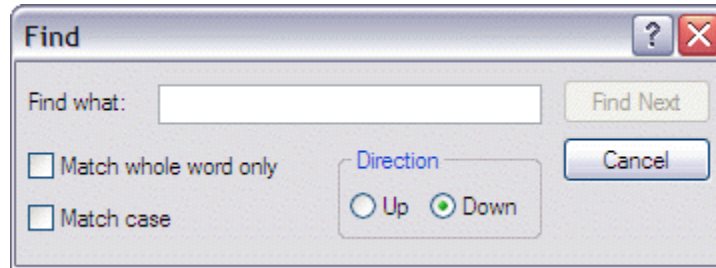
- 6 Repeat [steps 1-2](#).

NOTE: The Top SQL Display for DB2 also includes the Dynamic SQL Cache tab, which displays cached dynamic SQL statements. On this tab you can use the [Find button](#) to search within the SQL list, just as you would on the Results tab. As well, you can use the [Explain SQL](#) button to run an explain plan on the selected SQL.

Completing the Find dialog in the Top SQL View

You can specify a search string to help you find top SQL in a results set.

- 1 On the **Results** tab of the **Top SQL** view, click **Find**.



- 2 In the **Find what** box, enter the search string.
- 3 Select any of the options to narrow the search and click **Find Next** to search for the target search string.

Running an Explain Plan/Query Plan on SQL Statements

Performance Center lets you run EXPLAIN/QUERY plans on SQL statements in a results set.

- 1 On the **Results** tab of the **Top SQL** view, in the upper grid, select an SQL statement.
- 2 Click **Explain SQL**. The target SQL statement is displayed in the lower grid, an expanded plan is run on the target SQL statement, and then lists the results in the lower grid.

NOTE: You must have the appropriate privileges to run an EXPLAIN/QUERY plan.

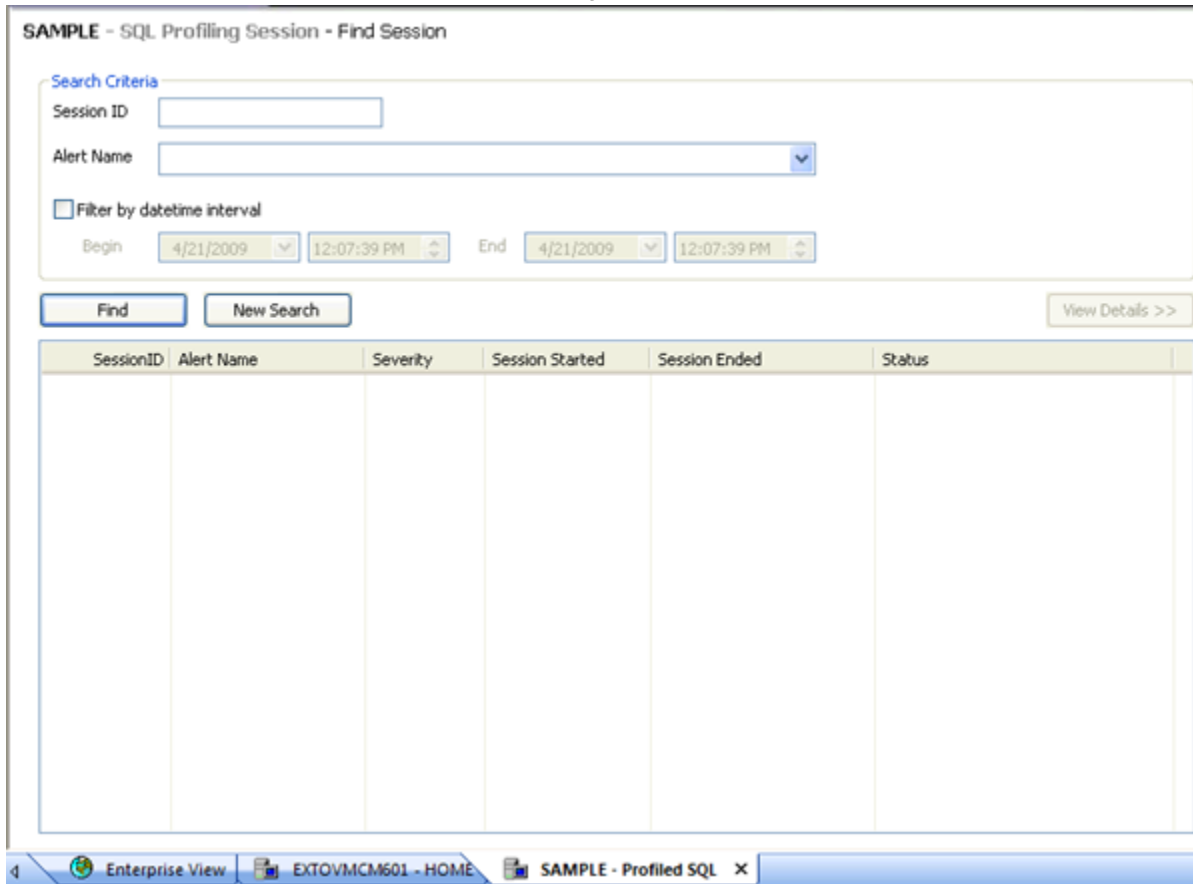
Profiled SQL

The Profiled SQL view allows you to search for SQL captures running or executed in the current Oracle, SQL Server, DB2 or Sybase datasource. When a violation occurs, a SQL Capture session is triggered and collects performance-related data for the SQL running currently of the datasource. The table below describes the functions available on the tabs of the Profiled SQL view:

Field	Description
Session ID	Enter the Session ID as a search criteria.
Alert Name	Click the drop-down arrow to choose the Alert Name you want to search for.
Filter by date/time interval	Click the Filter by datetime interval to enter a search interval based on date and time.
Find	Click Find to begin the search.
New Search	Click New Search to clear the search criteria parameters and enter new criteria.
View Details	Once the search is complete, the View Details button is available.

Opening the Profiled SQL View

- 1 In the Registered Datasources tree expand the Datasource to view the Profiling SQL node.
- 2 Double-click **Profiled SQL** to open the SQL Profiling Session view



Using the View Details Option

Once a search in the Profiled SQL view is complete, the View Details button is available.

- 1 Select one of the results listed in the search view and click **View Details**.
- 2 The SQL Profiling Session view displays the following information:
 - Registered Datasource name
 - Alert Name what was the search criteria
 - Date and time the alert was generated
 - Session ID
 - Details of the alert
 - Stored procedure of the alert

Trends

The Trends view is available to let you simultaneously track statistics for monitored datasources. This is especially useful for determining enterprise trends. The Trends view offers the following functionality:

- [Add](#)
- [Delete](#)
- Open
- Save

NOTE: For detailed information on the first statistic in the grid, right-click the chart, and then click Help.

For more information, see:

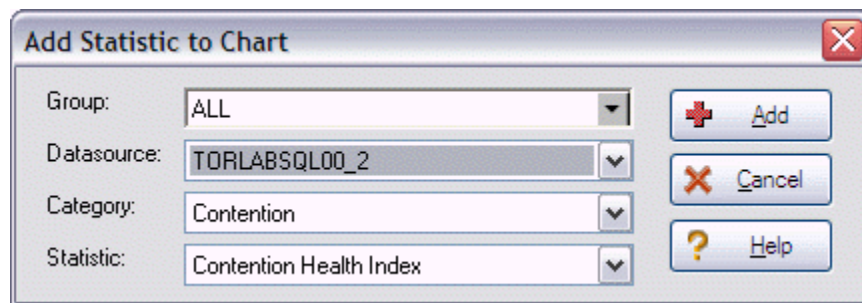
[Completing the Add Statistic to Chart dialog](#)

[Deleting a Live Statistic from the Trends Chart](#)

Completing the Add Statistic to Chart dialog

Performance Center lets you select statistics to include in the trends graph.

- 1 Click **Trends >Add** and the Add Statistic to the Chart dialog. appears



- 2 Click **Group**, and then click the target statistic's datasource group.
- 3 Click **Datasource**, and then click the target statistic's datasource.
- 4 Click **Category**, and then click the target statistic's performance category.
- 5 Click **Statistic**, and then click the target statistic.

NOTE: For detailed information on the target statistic, click Help.

- 6 Click **Add** and the live statistic is added to the chart.
- 7 To add more statistics to the chart, repeat [steps 2-6](#).
- 8 When you finish adding live statistics to the chart, click **Close**.

Deleting a Live Statistic from the Trends Chart

Performance Center lets you select statistics to delete from the trends graph.

- Click **Trends > Delete** and the statistic from the Trends chart is deleted.

SQL Scripting

Performance Center incorporates a powerful SQL scripting environment, the ISQL Window. The ISQL Window lets you write, debug, test, and deploy solid SQL code for your database applications. The scripting environment lets you:

- Open multiple interactive script windows.
- Execute multiple scripts on the same desktop.
- Capture multiple result sets on the same desktop.

Performance Center's scripting environment is comprised of different windows:

- [ISQL Window](#)
- [Results Editor](#)

These windows are context sensitive to the type of script you are opening or extracting. For example, if you extract the schema for a table, Performance Center opens a DDL Window containing the script. If you execute a script, a result window displays containing the results of your execution statement.

TIP: Since you must drop a database object before you can recreate it, you can set the DDL Editor to automatically include DROP statements for specified objects.

TIP: You can have multiple ISQL windows open at the same time, with each running separate queries.

SQL Editor

The ISQL Window lets you:

- [Open ISQL Windows](#)
- [Open Script Files](#)
- [Find and Replace in an ISQL Window](#)
- [Use Regular Expressions in ISQL Windows](#)
- [Navigate in an ISQL Window](#)
- [Save and Close Scripts](#)

ISQL Windows

A script is a collection of SQL statements used to perform a task. Scripts, stored as files, form the foundation of most application projects. Definitions of tables, indexes, stored procedures, as well as reports and batch jobs, are usually stored in script files and shared by groups of developers using version control software. You can manage SQL scripts in ISQL Windows. Performance Center lets you open multiple SQL windows in a Workbook.

Opening ISQL Windows

To open an ISQL Window, do the following:

- 1 On the main menu click **Datasource > SQL > New**.

OR

On the toolbar, click **New SQL** (). An ISQL window opens as a tabbed view.

For more information, see [ISQL Editor](#).

Open Files

The Open Files dialog lets you open existing files.

Important Notes

- None

The table below describes the options and functionality on the Open Files dialog:

Option	Description
File Name	Type a script file name. You can open multiple script files simultaneously by selecting multiple file names in the file list.
Files of Type	Control the files displayed in the file list by changing the selection. Performance Center displays multiple default file extensions.
Open Button	Click to open one or more files into the current workspace.

Completing the Open Files dialog

To complete the Open Files dialog, do the following:

- 1 On the main menu click **Datasource > SQL > Open** and the Open Files dialog appears.
- 2 In the **Open Files** dialog, do one of the following:
 Select a script file.
 OR
 In **File Name**, type a script file name, and then in the **Files of type** box, select types of files to display.
- 3 Click **Open** and the selected script opens in the ISQL Editor.

For more information, see [ISQL Editor](#).

Find

The Find dialog lets you search text in your SQL scripts.

Important Notes

- None

The table below describes the options and functionality on the Find dialog:

Option	Description
Find What	Type your search string.
Match whole word only	Select to search for only the complete word.
Match Case	Select to make the search case-sensitive.
Regular expression	Select if you are looking for a regular text expression .
Wrap around search	Search from the end of the script and back to the insertion point.

Option	Description
Direction	Specify the direction you want to search. Click the Up or Down option button.
Find Next Button	Click to find the next occurrence of your search string.
Mark All Button	Click to place a small blue dot next to every line number in the script which meets the required search string criteria.

For more information, see [Completing the Find dialog](#).

Completing the Find dialog

To complete the Find dialog, do the following

- 1 On the main menu click **Edit > Find**.
- 2 In **Find What**, type your search string.
- 3 Select the options you want for the find function:
 - Select **Match whole word only** to search for only the complete word.
 - Select **Match Case** to make the search case-sensitive.
 - Select **Regular expression** to search for a regular text expression.
 - Select **Wrap around search** to search from the end of the script and back to the insertion point.
- 4 In **Direction**, click **Up** or **Down**.
- 5 Click **Find Next** to find the next occurrence of your search string.
- 6 Click **Mark All** to place a small blue dot next to every line number in the script which meets the required search string criteria and the ISQL Editor highlights the object name if it matches the search criteria.

For more information, see [Find](#).

Replace

The Replace dialog lets you search and replace text in your SQL scripts. The table below describes the options and functionality on the Replace dialog:

Option	Description
Find What	Type your search string.
Replace With	Type the replacement text.
Match whole word only	Select to make the search for the whole word only and not words that are part of a larger word.
Match Case	Select to make the search case-sensitive.
Regular expression	Select if you are looking for a regular text expression .
Wrap around search	Search from the end of the script and back to the insertion point.
Direction	Specify the direction you want to search. Click the Up or Down option button.
Find Next Button	Click to find the next occurrence of your search string.
Replace Button	Click replace the current selection.
Replace All Button	Click to automatically find and replace all occurrences of your search string within the current window.

For more information, see [Completing the Replace dialog](#).

Completing the Replace dialog

To complete the Replace dialog, do the following:

- 1 On the main menu click **Edit > Replace**.
- 2 In **Find What**, type your search string.
- 3 In **Replace With**, type the replacement text.
- 4 Select the options you want for the replace function:
 - Select **Match whole word only** to search for only the complete word.
 - Select **Match Case** to make the search case-sensitive.
 - Select **Regular expression** to search for a regular text expression.
 - Select **Wrap around search** to search from the end of the script and back to the insertion point.
- 5 In **Direction**, click **Up** or **Down**.
- 6 Click **Find Next** to find the next occurrence of your search string.
- 7 Click **Replace** to replace the current selection.
- 8 Click **Replace All** to automatically find and replace all occurrences of your search string within the current window.

For more information, see [ISQL Editor](#).

Regular Expressions

Regular Expressions are offered as an optional search criteria in the SQL windows search facility. Regular Expressions serve as powerful notation for describing string matching patterns. Special characters are used to denote certain match criteria on which the ISQL Editor should conduct its search. The table below describes the special characters and their meanings:

Character	Meaning	Example
^	Circumflex - Constrains the search to the start of a line	^Rap -- Match lines beginning with Rap ^Emb -- Match lines beginning with Emb
\$	Dollar - A dollar as the last character of the string constrains matches to the end of lines.	if\$--Match lines ending with if ^end\$-- Match lines consisting of the single word end
.	Period - A period anywhere in the string matches any single character.	T..l -- Matches Tool, Till, Tail etc. H.w -- Matches Huw, How, Haw etc. ^Sin.ers -- Matches lines beginning with Sinders, Sinners etc.
*	Asterisk - An expression followed by an asterisk matches zero or more occurrences of that expression.	to* -- Matches t, to, too etc. 00* -- matches 0, 00, 000, 0000 etc.
+	Plus - An expression followed by a plus sign matches one or more occurrences of that expression.	to+ -- Matches to, too etc. 10+ -- Matches 10, 100, 1000, 10000 etc. /(:(d+)/ -- Matches (0), (12464), (12) etc.

Character	Meaning	Example
?	Question mark - An expression followed by a question mark optionally matches that expression.	for? -- Matches f and for 10? -- Matches 1 and 10
()	Brackets - A set of brackets can be used to group characters together prior to using a * + or?.	Rap(id)? -- Matches Rap and Rapid B(an)*a -- Matches Ba, Bana and Banana
[]	Square brackets - A string enclosed in square brackets matches any character in that string, but no others. If the first character of the string is a circumflex, the expression matches any character except the characters in the string. A range of characters can be specified by two characters separated by a -. These should be given in ASCII order (A-Z, a-z, 0-9 etc.).	{[0-9]} -- Matches {0}, {4}, {5} etc. /[([0-9]+)/] -- Matches (100), (342), (4), (23456) etc. H[uo]w -- Matches Huw and How Gre[^\py] -- Matches Green, Great etc. but not Grep, Grey etc. [z-a] -- Matches nothing ^[A-Z] -- Match lines beginning with an upper-case letter
\	Backslash - A backslash quotes any character. This allows a search for a character that is usually a regular expression specifier.	\\$ -- Matches a dollar sign \$ \+ -- Matches a +

For more information, see [ISQL Editor](#).

Goto

In the Goto dialog you can move to a specific line in your script.

Important Notes

- None

The table below describes the options on the Goto dialog:

Option	Description
Line Number	Type or select the target line number.

Completing the Goto dialog

To complete the Goto dialog, do the following:

- 1 On the main menu click **Edit > Goto**.
- 2 In **Line Number**, type or select the target line number and click **OK** and the cursor moves to the target line.

For more information, see [ISQL Editor](#).

Saving and Closing Scripts

Untitled scripts are named SQL1...SQLn by default, and all scripts are saved with the SQL extension. You can save your files with any valid name.

For more information, see:

[Saving a Script](#)

[Closing a Script](#)

[Closing the Error List](#)

[ISQL Editor](#)

Saving a Script

To complete the Save dialog, do the following:

On the main menu click **Datasource > SQL > Save**.

OR

On the **ISQL window** toolbar, click  and the Save As dialog appears.

- 3 If this is a new file, enter the name of the file in **File Name**.
- 4 If this is an existing file and you want to use save as, on the main menu click **Save As**.
- 5 In the **File Name** box, type the name of the file.
- 6 Click **Save** and the Save As dialog closes.

For more information, see [Saving and Closing Scripts](#).

Closing a Script

To close a script, do the following:

- 1 On the **ISQL window** tool bar, click **Close**.

OR


Right-click in the **ISQL Editor**, and then click **Close**.

- 2 If you have not saved your script, you are prompted to save the file. Click **Yes** to save and **No** to close without saving.

For more information, see [Saving and Closing Scripts](#).

Closing the Error List

To close the error list, do the following:

- 1 On the ISQL window tool bar, click Errors ()


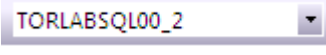




OR

Right-click in the **ISQL Editor**, and click **View Error**

For more information, see [ISQL Editor](#).

Editing Scripts

The ISQL Window incorporates a powerful editor that lets you edit your SQL scripts. The table below describes the commands found on the ISQL Editor Toolbar:

Button	Command	Description
	Execute	Runs the script or selected SQL.
	Select Datasource	List where you can specify the datasource against which to execute the SQL.
	Select Database	Specify the database against which to execute the SQL.
	Errors	Toggles the display of the Error window below the ISQL window.
	Save	Opens the Save As dialog, where you can save the SQL as a file.
	Close	Closes the ISQL window.

Editing a Script

To edit a script, do the following:

- 1 In the **SQL** window, type your changes or additions.
- 2 When you are finished with your changes, on the **ISQL** window tool bar, click [Execute](#) to compile the script.

For more information, see [ISQL Editor](#).

Selecting Text

You can select a single word, a single line, or a block of text.

Selecting a Single Word

To select a single word, do the following:

- 1 In the **ISQL Editor** window, position the pointer in the word and double-click.

Selecting a Line of Text

Performance Center offers two ways to select a line of text:

- 1 In the **ISQL Editor** window, click the line number listed on the left side of the window.
OR
- 2 Position the pointer at the beginning of the line of text, hold down the SHIFT key, and then click the end of the line of text.

Selecting a Block of Text

To select a block of text, do the following:

- 1 In the **ISQL Editor** window, drag until the block of text is selected.

For more information, see [Editing Scripts](#).

Cutting, Copying and Pasting Text and Columns in an ISQL Window

The Editor window incorporates Cut, Copy and Paste text functions. You can move the selected text to and from the Windows clipboard.

For more information, see:

[Copying and Pasting Text](#)

[Cutting and Pasting Text](#)

Copying and Pasting Text

To copy and paste text, do the following:

- 1 In the **Editor** window, select the target text.
- 2 Copy:
On the **Edit** menu, click **Copy**.
OR
Right-click in the **ISQL Editor**, and then click **Copy**.
- 3 Paste:
On the **Edit** menu, click **Paste**.
OR
Right-click in the **ISQL Editor**, and then click **Paste**.

Cutting and Pasting Text

To cut and paste text, do the following:

- 1 In the **Editor** window, select the target text.
- 2 Cut:
On the **Edit** menu, click **Cut**.
OR
Right-click in the **ISQL Editor**, and then click **Cut**.
- 3 Paste:
On the **Edit** menu, click **Paste**.
OR
Right-click in the **ISQL Editor**, and then click **Paste**.

Setting Bookmarks

Bookmarks are useful tools for navigating throughout an Editor window. You can jump back and forth between bookmarks easily, and there is no practical limit to the number of bookmarks you can set.

The table below describes the Edit menu commands for setting bookmarks:

Feature	Description
Toggle Bookmark	Set or clear a bookmark in the gutter next to a line of interest.
Previous Bookmark	Go back to the closest bookmark that is above your current location.
Next Bookmark	Go to the next bookmark below your current location.
Clear All Bookmarks	Clear every bookmark from the script file.

Setting a Bookmark

To set a bookmark, do the following:

- 1 In the **Editor** window, position the pointer in front of the line you want to bookmark.
- 2 On the **Edit** menu, click **Toggle Bookmark** and a blue dot is inserted in the gutter next to the line you have bookmarked.

For more information, see:

[Moving Between Bookmarks](#)

[Clearing Bookmarks](#)

Moving Between Bookmarks

You can use the Next Bookmark and the Previous Bookmark commands to move back and forth between bookmarks.

Goto the Next Bookmark

To goto the next bookmark, do the following:

- 1 In the **Editor** window, position the pointer where you have set bookmarks, and then on the **Edit** menu, click **Next Bookmark**.

Goto the Previous Bookmark

To goto the previous bookmark, do the following:

- 1 In the **Editor** window, position the pointer where you have set bookmarks, and then on the **Edit** menu, click **Previous Bookmark**.

For more information, see [Setting Bookmarks](#).

Clearing Bookmarks

To clear bookmarks, do the following:

- 1 On the **Edit** menu, click **Clear All Bookmarks**.

NOTE: This does not clear any error indicators (red dots) that might be showing in the gutter.

For more information, see [Setting Bookmarks](#).


Executing Scripts

The ISQL Editor lets you execute all or part of your SQL scripts. Unless you are executing large scripts that have multiple statements within them, or you need to view optimizer statistics, you execute most of your SQL scripts without options.

Executing a Script

To execute a script, do the following:

- 1 To execute a script:

On the **Editor** window tool bar, click  .

OR

In the **Editor** window, right-click, and then click **Execute**.

TIP: Performance Center lets you execute partial scripts too. Simply highlight the target portion of the script before clicking the Execute button.

Canceling a Query

The ISQL Editor lets you cancel a query while the rows are still being returned.

Canceling a Query

To cancel a query, do the following:

- 1 On the **Editor** window tool bar, click the **Cancel** icon.

NOTE: This button is only enabled after a script has begun executing.

For more information, see [Executing Scripts](#).

Results Editor

The results of your SQL queries are displayed in the Results tab of each Editor Window, which captures result sets in a virtual data grid that accommodates large result sets. The data grid offers many of the features of a basic Windows spreadsheet, giving you a great deal of flexibility in editing the worksheet and formatting its contents.

You have many choices for navigating and viewing your SQL query results. The Results window includes functionality to set result window options, find and replace, export data to other products such as Microsoft Excel, and mail your results files.

Result Window Status Bar

The Result Window Status Bar displays information about each Result window at the bottom of each window. You can display or hide the status bar by setting the Status Bar preference. This preference applies to all result windows.

For more information, see [Results Editor](#).

Closing Result Window Tabs

You can close tabbed Result set windows.

Renaming a Result Window Tab

To rename a result window tab, do the following:

- 1 Place your pointer over the **Results** tab.
- 2 Right-click the **Results** tab, and then click **Rename**.
- 3 In the **New Name** box, type the new name.
- 4 Click **OK** and the name is changed and the Rename Tab Editor is closed.


For more information, see [Results Editor](#).

Saving and Closing Result Sets

You can save your result sets using the standard Save and Save As functions. You can save multiple copies of the same result set and modify each copy to specific formatting requirements.

Saving Results

To save results, do the following:

- 1 Open the **Save As** dialog:
 - On the **Datasource** menu, point to **SQL**, and then click **Save**.
 - OR
 - On the **Editor** toolbar, click  .
- 2 In the **File name** box, type the name of the result set.
 - NOTE:** By default, the product appends the rsl extension to result set files. You can change the extension if you like, but the file is always treated as a result file.
- 3 To include column titles, select **Include column titles when saving**.
- 4 Click **Save**.

Closing a Result Set

To close a result set, do the following:

- 1 Close a Result Set:
 - On the **Datasource** menu, point to **SQL**, and then click **Close**.
 - OR
 - On the **Result** tool bar, click the **Close** icon.
 - OR
 - Right-click the **Results** window, and then click **Close**.
- 2 If you have not saved your result set, Performance Center prompts you to save the file. Click **Yes** to save or **No** to close without saving.

For more information, see [Results Editor](#).

Editing Result Sets

The Results Editor provides many ways to edit and customize your result windows. The Data Grid offers a host of features for manipulating, sorting and formatting data.

For more information, see:

[Cutting, Copying, and Pasting Cell Contents](#)

[Cutting, Copying, and Pasting Rows](#)

[Cutting, Copying, and Pasting Columns](#)

[Resizing Rows and Columns](#)

[Sorting Data](#)

Cutting, Copying, and Pasting Cell Contents

The Result window supports standard cut, copy and paste functionality.

Cutting Cell Contents

To cut cell contents, do the following:

- 1 In the **Results** window, double click or tab to the target cell. A double line bounds the selected cell. You can also select text using standard text selection techniques.
- 2 On the **Edit** menu, click **Cut** and the cell is cut.

Copying Cell Contents

To copy cell contents, do the following:

- 1 In the **Results** window, double click or tab to the target cell. A double line bounds the selected cell. You can also select text using standard text selection techniques.
- 2 On the **Edit** menu, click **Copy** and the cell is copied.

Pasting Cell Contents

To paste cell contents, do the following:

- 1 In the **Results** window, double click or tab to the target cell. A double line bounds the selected cell.
- 2 On the **Edit** menu, click **Paste** and the cell is pasted.

For more information, see [Editing Result Sets](#).

Cutting, Copying, and Pasting Rows

You can perform standard cut, copy, and paste functions on rows, just as you can on individual cells.

Cutting Rows

To cut a row, do the following:

- 1 In the **Results** window, click the numbered row heading on the left side of the row.
- 2 On the **Edit** menu, click **Cut** and the row is cut.

Copying Rows

To copy rows, do the following:

- 1 In the **Results** window, double-click or tab to the target row. A double line bounds the selected row. You can also select text using standard text selection techniques.
- 2 On the **Edit** menu, click **Copy** and the row is copied.

Pasting Rows

To paste rows, do the following:

- 1 In the **Results** window, double-click or tab to the target row. A double line bounds the selected row.
- 2 On the **Edit** menu, click **Paste** and the row is pasted.

For more information, see [Editing Result Sets](#).

Cutting, Copying, and Pasting Columns

You can perform standard cut, copy, and paste functions on columns, just as you can on rows.

Cutting Columns

To cut columns, do the following:

- 1 In the **Results** window, click the column heading above the first row.
- 2 On the **Edit** menu, click **Cut** and the column is cut.

Copying Columns

To copy columns, do the following:

- 1 In the **Results** window, click the column heading.
- 2 On the **Edit** menu, click **Copy** and the column is copied.

Pasting Columns

To paste columns, do the following:

- 1 In the **Results** window, click the column heading above the first row to select the target column.
- 2 On the **Edit** menu, click **Paste** and the column is pasted.

For more information, see [Editing Result Sets](#).

Resizing Rows and Columns

Resizing rows and columns can aid readability of the spreadsheet contents or condense space for editing and formatting purposes.

Resizing Rows to Their Default Height

To resize rows to their default height, do the following:

- 1 Select one or more rows by clicking the numbered row headings to the left of the rows.
- 2 Right-click the selected rows, and then click **Resize Rows**.

Resizing Rows to a New Height

To resize rows to a new height, do the following:

- 1 Select one or more rows by clicking the numbered row headings to the left of the rows.
- 2 Change the pointer to a solid horizontal bar with arrows on top and bottom by moving it to one of the upper or lower borders of the row heading.
- 3 Click and grab the row border, and then drag the pointer to enlarge or shrink the height of the row.

Resizing Columns to Their Default Widths

To resize columns to their default widths, do the following:

- 1 Select one or more columns by clicking the column headings.
- 2 Right-click the selected columns, and then click **Resize Columns**.

Resizing Columns to a New Width

To resize columns to a new width, do the following:

- 1 Select one or more columns by clicking the column headings.
- 2 Change the pointer to a solid horizontal bar with arrows on top and bottom by moving it to one of the upper or lower borders of the column heading.
- 3 Click and grab the column border and drag the pointer to enlarge or shrink the height of the column.

For more information, see [Editing Result Sets](#).

Sorting Data

To order and organize data in a coherent manner, you can sort columns alphanumerically in single result sets.

Sorting Data

To sort data, do the following:

- 1 In the data grid, select the column(s) you want to sort.
- 2 Double click the column header(s) to sort the data in the column in ascending, alphanumeric order. To sort the column in descending order, double click the column header again.

NOTE: This option is not valid for multiple result sets.

For more information, see [Editing Result Sets](#).

Formatting Result Sets

The ISQL Editor provides flexibility in formatting result sets, for analysis and reporting, from within a Result Window. Result sets can be formatted to best suit your purpose, whether it be sending via e-mail, printing, or exporting a file to other programs, such as Microsoft Excel. Some of these features change only the screen display of the results, while others allow you to format for printing. Performance Center includes the Format Styles and Display Settings dialogs where you can configure the Results Window display.

For more information, see:

[Setting Alignment Properties](#)

[Setting Border Properties](#)

[Setting Result Set Display Properties](#)

[Setting Fonts](#)

[Setting Colors](#)

Setting Alignment Properties in Result Windows

You can use the format menu or the shortcut menu to alter result set alignment properties. Selecting the Alignment command displays the Format Styles tabbed dialog:

Setting Alignment Properties

To set alignment properties, do the following:

- 1 Right-click the **Result** data grid, and then click **Alignment** and the Format Styles dialog appears.
- 2 Click the **Align** tab.
- 3 To change **Horizontal** properties, click the appropriate **option** button in the **Horizontal** box.
- 4 To enable **Wrap Text**, **Allow Enter** and/or **Auto Size**, select the appropriate options.
- 5 To disable **Wrap Text**, **Allow Enter** and/or **Auto Size**, select the appropriate options and deselect these options.

The table below describes alignment properties:

Property	Option	Description
Horizontal	Standard	Aligns data in cells based on their default datatype alignment properties. For example, numbers are right aligned, whereas text is left aligned.
	Left Aligned	Forces all data to be left aligned, regardless of datatype
	Center	Forces centering of all data, regardless of datatype
	Right Aligned	Forces all data to be right aligned, regardless of datatype
Vertical	Standard	Aligns data in cells based on their default datatype alignment properties. For example, numbers are right aligned, whereas text is left aligned.
	Top	Forces all data to be aligned to the top of each cell
	Center	Forces all data to be aligned in the center of each cell
	Bottom	Forces all data to be aligned at the bottom of each cell
Wrap Text		Specifies that data exceeding the length of the cell should wrap to additional lines in the same cell
Allow Enter		Specifies that the contents of cells can be edited or not
Auto Size		Specifies whether or not rows should size automatically to accommodate the longest data item in a column

For more information, see [Formatting Result Sets](#).

Format Border Styles

You can use the format menu or the shortcut menu to alter border properties. Selecting the Border command displays the Format Styles tabbed dialog.

Completing the Format Styles dialog

To complete the Format Styles dialog, do the following:

- 1 Right-click the **Result** data grid, and then click **Borders** and the Format Styles dialog appears.
- 2 Click the **Borders** tab.
- 3 On the **Border** box, you can indicate whether or not a border should appear on the top, bottom, right, left, or for a range of cells by clicking the corresponding boxes.
- 4 To set a range of cells apart by setting a particular border style around that range, select the range in the result set before opening the **Format Styles** dialog. To select the **Range** property, click the range box.
- 5 In the **Type** box, you can select the type of line you want to border the cell or cells by clicking the corresponding boxes.
- 6 To select a color, click the **Color** list, and then click the border color.
- 7 Click **OK**. Your changes are saved and the Format Styles dialog closes.

For more information, see [Formatting Result Sets](#).

Display Settings

You can use the format menu or the shortcut menu to alter display properties. Selecting the Properties command displays the Display Setting dialog.

The Display Settings dialog lets you set a number of format properties for the result set window:

- How titles and grid lines are displayed.
- How lines in the grid are displayed.
- How the result set window background is displayed.
- How current cells, rows and columns are indicated.

Completing the Display Settings dialog

To complete the Display Settings dialog, do the following:

- 1 Right-click the **Result** data grid, and then click **Properties** and the Display Settings dialog appears.
- 2 In the **Titles and Gridlines** box, select the target display properties options. The change is displayed in the right-pane **Preview** window of the **Properties** dialog.
- 3 **In the Color** box, click the property you want to change, and then click the new color. Changes are propagated automatically when you change them.
- 4 To alter the user properties, in the **User Properties** list, click the attribute you want applied to the current cell.
- 5 To save your changes to the current configuration, select the **Save settings to profile** option.
- 6 Click **OK**. Your changes are saved and the Properties dialog is closed.

For more information, see [Formatting Result Sets](#).

Format Font Styles

You can use the format menu or the shortcut menu to alter font properties. Selecting the Font command displays the Format Styles tabbed dialog.

Completing the Format Styles dialog

To complete the Format Styles dialog, do the following:

- 1 Right-click the **Result** data grid, and then click **Font** to open the Format Styles dialog.
- 2 Click the **Font** tab.
- 3 In the **Font Combo** box, type or click the font you want to use.
- 4 In the **Outline Combo** box, type or click the outline you want to use.
- 5 In the **Size Combo** box, type or click the size you want to use.
- 6 To make a line cross through the length of the text, in the **Effects** box, select the **Strikeout** option.
- 7 To underline the text, in the **Effects** box, select the **Underline** option.
- 8 To change the text color, click the **Text Color** list, and then click the new color. The Preview box displays the sample text of your selections

For more information, see [Formatting Result Sets](#).

Format Color Styles

You can use the format menu or the shortcut menu to alter color properties. Selecting the Color command displays the Format Styles tabbed dialog.

The table below describes the options and functionality on the Format Styles dialog:

Option	Description
Interior	To change the result set pattern, in the box, click the list, and then click a new pattern. To set the foreground color of the result set, in the box, click a foreground group color button. To set the background color of the result set, in the box, click a background group color button.
3D-Effect	To change the 3D-Effect, in the box, click the appropriate option button: The Normal option button provides a standard flat appearance. The Raised option button provides a raised 3D appearance in the selected cells. The Inset option button provides a depressed appearance in the selected cells. The default is typically set to Normal.

Completing the Format Styles dialog

To complete the Format Styles dialog, do the following:

- 1 Right-click the **Result** data grid, and then click **Color** and the Format Styles dialog appears.

For more information, see [Formatting Result Sets](#).

Performance Center Web Client

Performance Center offers a Web Client so DBAs and operators can review database performance statistics from any Web browser without requiring any additional software. The Web Client is automatically installed on the server machine and can be accessed from any client or from a remote computer when you type in the URL as long as the Performance Center is running.

The statistics you see from the “fat client” are all visible in the Web Client with the exception of Oracle’s OS page statistics, so you can how many, and where, alarms have been triggered. You can also easily review drill-down details by clicking a target statistic.

NOTE: Performance Center Web Client is also fully integrated into DBArtisan, which means that a DBA can view the new Web Client from within DBArtisan and monitor all the datasources that are registered with Performance Center.

To access the Web Client from a Performance Center client or server machine, on the main menu click **View>Web Client**. The Web Client opens as a tabbed view in the Performance Center application.

You can also click **Start > Programs > Embarcadero Performance Center > Web Client**. The Web Client opens as a separate browser.

To access the Web Client remotely, type the URL, which would look like:

`<http://SERVERMACHINE/webclient>`

or

`<http://SERVERMACHINE/perfcntr/ServerManager.dll?RequestPage?Page=EnterpriseView>`.

From the Web Client, you can see:

- [Enterprise View](#)
- [Monitor Details](#)
- [Server Details](#)
- [Client Details](#)
- [Admin Login](#)
- [Reports](#)
- [Help](#)

Enterprise View

The Enterprise view is a single view designed to give you a comprehensive picture of the overall health of all monitored datasources in your enterprise. It displays all currently registered datasources. The Enterprise view displays what databases are up and down and gives you other critical indicators regarding the status of your monitored datasources. The Enterprise view for the main or “fat” client and the Web client differ slightly:

The table below describes the Enterprise View columns for the Web Client:

Column	Description
Datasource	Displays the name of the datasource(s).
DBMS	Displays the datasource’s platform.

Column	Description
Version	Displays the platform's version for each datasource.
Datasource Up	Indicates if the datasource is running.
Listener Up	ORACLE ONLY: Indicates if the Oracle-specific datasource listener is functioning.
Health Index	Indicates the current overall performance level of the registered datasource. The Health Index is on a scale of 0-100 with 100 being the ideal.
Alarms	Indicates any alarms for the registered datasource.

Enterprise View Graphics

In addition to the information in the columns, the Enterprise view colors each graphic, for both the main Enterprise View and the Web client, to give you a general overview of the health of your datasources and to alert you to any threshold violations. The table below describes the color and associated severity level for your datasources:

Color	Severity
Green	Low
Yellow	Warning
Orange	High
Red	Critical

You can use the Enterprise view to quickly determine the status of the registered datasources in your enterprise.

Performance Center Web Client Views

The performance category views for the Web Client provide you with important statistics for monitoring your datasources. The overall categories are the same for all supported platforms, however, the information available is specific to the datasource currently selected. You can open more than one performance category view by selecting a datasource from the [Enterprise](#) view.

When you click a target datasource, you see the same statistics that you see from the "fat client." You can also click to see the drill-down details for a target metric.

In addition to the general Enterprise View, the Web Client offers a [Monitor Details](#) view, a [Server Details](#) view, a [Client Details](#) view, an [Admin Login](#) view, and a [Reports](#) view.

Monitor Details

In the Monitor Details view of the Web client you can see performance details of all registered datasources at a glance. The table below describes the information available to you from this view:

Column	Description
Monitored Datasource	Displays the name of the datasources registered to the server you are monitoring.
Monitor	Show the monitoring capability and whether or not the datasource is fully enabled or disabled. You enable or disable datasource monitoring from the Datasource Properties dialog.
Status	Shows the current datasource status and whether or not it is OK. Other status possibilities are Error, Unknown, or Transition, but these are rarely, if ever seen.

Column	Description
Connection Detail	Displays the status of the connection of the datasource to your server. Ideally it is connected and deployed. Other possibilities are Connected/Deployed Failed, Not Connected, Connected/Deployed Execution Error, and so on.
User ID	The user ID.
Query Timeout (seconds)	Shows the amount of time, in seconds, before a query times out. The default is 300.
Ping Interval (seconds)	Shows the amount of time between pings from the server to the datasource. The default is 300.
Primary Interval (seconds)	Displays the amount of time between primary data sampling intervals. The default is 300
Secondary Interval (seconds)	This options displays the amount of time between secondary data fetch intervals. The default is 900 seconds.
Last Ping Time	Displays the date and time the datasource was most recently pinged.
Last Primary Fetch Time	Displays the date and time, and the amount of time it took to fetch the primary data most recently.
Last Secondary Fetch Time (seconds)	Displays the date and time, and the amount of time it took to fetch the secondary data most recently.
DBMS	Shows the database platform for the target datasource.
Version	Shows the database platform version for the target datasource.

Server Details

In the Server Details view of the Web Client you can see information about the server you are currently connected to.

TIP: To connect to another server, you can use the Select Performance Center Server dialog or you can type the server name in the Web Client url:

`<http://pc_host/perfctr?ServerManager.dll?RequestPage?Page=default>`

where "pc_host" equals the name or IP of the machine hosting the Performance Center server. You can open multiple windows to see details for more than one server simultaneously.

NOTE: To start or stop the target server from this page, go to the [Admin Login](#) view and enable your administrator status.

The table below describes the information available from this view:

Column	Description
Server Name	Display see the name of the server you are currently connected to.
Port	Displays the server's port number. The default is 80.
Web Server	Displays the web server. Currently Apache 2.6.2 is available.
Status	Shows the server's status. The options are OK, Error, Unknown, or Transition.
Status Detail	Shows whether the server is started or not.
Jobs	Displays the number of jobs currently running on the server.
Repository Datasource	Displays the name of the datasource where the repository is stored.
Status	Displays the repository status and whether it is OK or not.
Connection Detail	Shows the connection status of the repository.

Column	Description
User ID	Displays the name of the repository.
DBMS	Displays the database platform for the repository.
Version	Displays the database platform version of the repository.

Client Details

The Client Details view of the Web Client gives you information about the client machine you are currently connected to. The table below describes the information you see:

Column	Description
Client ID	This is the number of times users launch the fat client.
Machine Name	The name of the client machine. In those cases where the server and the client are on the same machine, you will see localhost.
IP Address	The IP address of the client machine.
Jobs	The number of jobs running on the client machine.
Msgs In	The number of messages sent from the server to the client.
Msgs Out	The number of messages sent by the client to the server is displayed.
Msg Queue Length	The number of messages waiting in the queue is displayed.
Msgs Lost	Shows see how many messages were lost. Client can lose messages because the Performance Center Server is producing too much data. If this happens, you can decrease the Polling Frequency to 2 or 1 second(s). However, decreasing the Polling Frequency causes the Performance Center Client to consume more CPU.

Admin Login

In the Admin Login, you can sign in as an administrator, if that status is available to you. Enter your user name and password and click **Authorize** to enable Administrator Status. You can also change the refresh rate settings. The default setting is 60 seconds.

When you are logged in as an administrator, you can start or stop the server from the Server Details page.

Reports

In the Reports function on the Web Client, you can generate reports remotely for data you have previously exported as a data extraction job. No special permissions are required. This gives you the opportunity to view reports wherever you have access to the Web Client.

It is important to keep this in mind when scheduling a data extraction job. If you are planning to run reports from a remote client, be aware that you cannot change the particulars of an extraction job from a distance. Also, the more data that's involved in the extraction job, the longer it takes to generate a report. From the fat client, you can always edit a data extraction job to focus on a smaller number of statistics or on a narrower time frame to ensure that generating a report will be less time-consuming and focused on areas of greatest interest to you.

NOTE: After you created a data extraction job, you can always get the most recent data by checking the refresh data box before you generate a report.

To Create or Rerun Report:

- 1 Click **Create Custom Report**.
- 2 Select the **Report Type**: You can choose Database Availability, Statistical Trend Analysis, or Alarms.
- 3 Select the **Job Name**: A list of the Data Extraction jobs you previously created appear.
- 4 To refresh data before you generate the report, check **Refresh Job Data**.
- 5 Enter a title in the **Report Title** box.

NOTE: You can also add a description of the report.

- 6 Click **Generate**.

NOTE: It can take a while for the report to generate, depending on how much data is involved. When the report is completed, the main report page appears. There's no overt indicator that the report is being generated, so you have to have faith that a report will come your way.

Reading Reports

After a report is generated, or when you open the Reports page, you see a list of file names and the date each report was created. Each report is available in three file formats:

csv,	The .csv (comma-separated variables) file displays each record as a single line. If you have Microsoft Excel on your client, the file will open in an Excel spreadsheet. You may need to increase column widths to see the data.
htm	The .htm file presents an overview of the statistics.
data.htm	The data.htm file gives you a drill-down view of each of the statistics.

NOTE: If you do not have Microsoft Excel, you can open the file using Notepad or Wordpad.

Reports you create in the Web Client are automatically saved to the fat client in the PC's root directory/ReportOutput/. You can change that directory path in the Server Options editor. To open the Server Options Editor, from the fat client's toolbar, go to **Monitor > Server Options > Report**.

Help

Click this option to open the Web Client help in a separate browser.

Microsoft SQL Server Expert Guide

This section includes expert help for all Microsoft SQL Server statistics in the Performance Center views:

- [Home View Statistics](#)
- [Database Page Statistics](#)
- [Contention Statistics](#)
- [I/O Page Statistics](#)
- [Memory Page Statistics](#)
- [Network Statistics](#)
- [OS Page Statistics](#)
- [Space Page Statistics](#)
- [Users Page Statistics](#)

Home View Statistics

The Performance Center Home view lets you review availability and overall performance of all monitored databases from a single window. Statistics on the Home view are organized into the following categories:

- [Contention Vital Signs](#)
- [I/O Vital Signs](#)
- [Memory Vital Signs](#)
- [Network Vital Signs](#)
- [Space Vital Signs](#)
- [Users Vital Signs](#)

Related Topics

[Database Page Statistics](#)

[Contention Statistics](#)

[I/O Page Statistics](#)

[Memory Page Statistics](#)

[Network Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

[Users Page Statistics](#)

Memory Vital Signs

The following memory statistics are on the SQL Server Home view:

- [Ad Hoc SQL Hit Ratio](#)
- [Buffer Cache Hit Ratio](#)
- [Log Cache Hit Ratio](#)
- [Procedure Plan Hit Ratio](#)

Buffer Cache Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

Data read from memory will produce end-user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the purposes of the SQL Server buffer/data cache.

The buffer cache hit ratio is a terrific indicator of how often user requests for data are satisfied through memory vs. being physically read from disk.

TIP: Click this statistic to drill down to the [Buffer Cache tab](#) of the Memory Detail view.

Metrics

To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, you should be aware that every server has its own 'personality' and might exhibit excellent performance with below average readings for the cache hit ratio. You should also be aware that excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

Consistently viewed low readings (60% or lower) might require tuning attention on the part of the DBA.

Troubleshooting

Ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server memory configuration option. Also, consider increasing your SQL Server Min. Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Often a user process is taking a large amount of memory due to an inordinate amount of I/O.

Procedure Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

The SQL Server procedure cache is used to hold the execution plans for all Transact-SQL statements currently executing in the server. When a user executes a Transact-SQL statement, SQL Server looks in the procedure cache for a query plan to use.

This statistic is the percentage of query plan requests generated by stored procedures that are found in the procedure cache area. The percentage of times that a statement's plan and definition can be referenced in memory, the better the procedure execution time.

TIP: Click this statistic to drill down to the [SQL Cache tab](#) of the Memory Detail view.

Metrics

A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for code reference. Note that when a database is first started, the procedure cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

Troubleshooting

First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Ad Hoc SQL Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

When an ad hoc SQL statement is issued, the query plan is then stored in the SQL Server procedure cache area. If the identical ad hoc statement is launched in the future, SQL Server uses the query plan already stored in the procedure cache if it is still there. The Ad Hoc SQL Hit Ratio statistic defines the percentage of times that a query plan for an ad hoc SQL statement is found in the procedure cache.

TIP: Click this statistic to drill down to the [SQL Cache tab](#) of the Memory Detail view.

Metrics

A high ad hoc hit rate is desirable, but is harder to maintain at a high level than something like a procedure cache hit rate. Therefore, an 80% or greater ad hoc cache hit rate is a good performance benchmark for code reference. Note that when a database is first started, the ad hoc cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the ad hoc cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

Troubleshooting

First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (To obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Log Cache Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

The Log Cache Hit Ratio represents the percentage of log cache reads satisfied from the log cache.

TIP: Click this statistic to drill down to the [Log Cache tab](#) of the Memory Detail view.

Metrics

None.

Troubleshooting

A low percentage on this statistic is not necessarily a bad sign, as it is possible that the information needed from the log will not be readily available in memory.

I/O Vital Signs

The following I/O statistics are on the SQL Server Home view:

- [I/O Errors](#)
- [Log Flushes](#)
- [Total Server Reads](#)
- [Total Server Writes](#)

TIP: Click any of these statistics to open the [I/O performance category view](#).

Total Server Reads

- [Metrics](#)

Total Server Reads reflect total number of physical reads performed by the database server since the last refresh inside Performance Center.

Metrics

Large numbers of physical reads could reflect a data or procedure cache that is too small. You should examine the data and procedure cache hit rates to determine the overall effectiveness of logical vs. physical I/O.

Total Server Writes

- [Metrics](#)

The Total Server Writes value reflects the total number of physical writes performed by the database server since the last refresh inside Performance Center.

Metrics

None.

Log Flushes

- [Metrics](#)

The Log Flushes statistic represents the total number of log pages for all databases written to disk by the log writer process. A log flush occurs when SQL Server writes all changes from the database's log cache out to the database's log files on disk.

Metrics

Increasing numbers observed for log flushes should not cause concern unless the I/O subsystem of the server appears overwhelmed. In addition, to minimize I/O contention between a database and its accompanying log, it is wise to place database files and log files on separate disks.

I/O Errors

- [Metrics](#)
- [Troubleshooting](#)

I/O Error Rate reflects total number of I/O errors (errors during read and write operations) encountered by the server since the last refresh inside Performance Center. I/O Error Rate is a percentage based on Total I/O (the sum the physical reads and writes).

Metrics

You should observe few, if any errors.

Troubleshooting

If you notice any errors, you should check the SQL Server error log for details.

Contention Vital Signs

The following contention statistics are on the SQL Server Home view:

- [Blocked Users](#)
- [Deadlocks](#)
- [Latch Waits](#)
- [Lock Timeouts](#)

Blocked Users

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely.

Metrics

You should immediately investigate any indicator above zero, before the situation has a chance to mushroom.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Then other user processes, almost always, complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into [Users Detail](#) and see all current blocking locks, learning exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Deadlocks

The Deadlocks statistic represents the number of deadlocks per second detected by SQL Server. Deadlocks occur when processes cannot proceed because they are waiting on a set of resources held by each other or held by other processes.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

Consistently seeing deadlock counts greater than zero will indicate that some user processes are experiencing delays in completing their work. When SQL Server identifies a deadlock, it resolves the situation by choosing the process that can break the deadlock. This process is termed the deadlock victim. SQL Server rolls back the deadlock victim's transaction, and then notifies the process' application by returning an error message. It also cancels the process' request and allows the transactions of the remaining processes to continue.

SQL Server always attempts to choose the least expensive thread running the transaction as the deadlock victim.

Troubleshooting

Because SQL Server automatically resolves deadlock situations, you should do proactive work to prevent them in the first place.

The culprit of most blocking lock and deadlock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

You can change default deadlock behavior by using the SET DEADLOCK_PRIORITY command, which reprioritizes a process' position in a deadlock situation.

Lock Timeouts

- [Metrics](#)

Number of lock requests per second that timed out, including internal requests for NOWAIT locks.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

None.

Latch Waits

- [Metrics](#)
- [Troubleshooting](#)

The Latch Waits statistic represents the number of latches per second that could not be satisfied immediately by SQL Server. Latches can be thought of as lightweight, mini-locks that are used to protect actions and resources used inside transactions.

TIP: Click this statistic to drill down to the [Waits tab](#) of the Contention Detail view.

Metrics

Latch waits rarely impact performance, however seeing latch waits with high wait time might indicate an area that needs further investigation.

Troubleshooting

Wait events can be hard to interpret at times. If you see a particular event that has caused a lot of wait time, you can review the information in this link (Microsoft Knowledge Base Article - 244455) to help understand the cause and potential remedy:

<http://support.microsoft.com/default.aspx?scid=kb:en-us:Q244455>

Space Vital Signs

The following space statistics are on the SQL Server Home view.

- [Databases Low on Space](#)
- [Logs Low on Space](#)

TIP: Click either of these statistics to open the [Space performance category view](#).

Databases Low on Space

A SQL server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The Databases Low on Space statistic indicates databases that have fallen below a recommended percentage of free space.

Metrics

This might or might not be a problem. Some databases are not dynamic in nature (meaning they are not expected to grow in size) and are sized so very little free space is present. However, growing databases are another situation and might require DBA intervention if their free space amounts run low.

Troubleshooting

If the percent used amount of a database is approaching problematic levels, there are three ways a DBA can rectify the situation:

- 1 The DBA can resize the current file(s) used by the database via an ALTER DATABASE ... MODIFY FILE command
- 2 The DBA can add a new file to the database via the ALTER DATABASE ... ADD FILE command.
- 3 The DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE ... MODIFY FILE ... FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

Of course, the DBA should also ensure that enough physical space exists on the server to accommodate additional database space.

Logs Low on Space

Each database in SQL Server has a transaction log, which is a serial record of all modifications that have occurred in a database as well as the transactions that caused each change. The Logs Low on Space statistic indicates transaction logs that have fallen below a recommended percentage of free space.

Metrics

If any log's used space exceeds the Performance Center recommended thresholds, then the DBA should take action to ensure that the log does not run out of available free space.

Troubleshooting

There are several things a DBA can do to ensure that a database's log does not run out of available free space:

- 1 First, most transactional-oriented databases should have their logs assigned to a separate physical drive than the database. Reasons for doing so include:
 - It prevents competition for space between the log and the database itself.
 - It allows the log to be monitored for space more effectively.
 - It improves performance.
- 2 If the database is not critical in nature, you can set the truncate log on checkpoint option `trunc log on chkpt`, which will eliminate any non-active space in the log when a database checkpoint occurs.
- 3 Critical databases needing higher levels of recovery should have schedules established that regular perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.
- 4 If a critical transaction log becomes full, it might be impossible to use standard procedures to dump transactions and reclaim space. The dump operation will likely have to incorporate the `no log` or `truncate only` options.
- 5 If a transaction log continuously approaches dangerously low levels of free space, then the DBA should allow the underlying file(s) of the log to automatically grow to meet the demand. This can be accomplished by using the `ALTER DATABASE ... MODIFY FILE ... FILEGROWTH` command. You should also ensure that the `MAXSIZE` setting for each file is set appropriately.

The DBA should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log since only non-active space in the transaction log is removed from log dumps or truncation operations.

Users Vital Signs

The following users statistics are on the SQL Server Home view:

- [Active Connections](#)
- [Current Locks](#)
- [Total Connections](#)
- [Transactions](#)

Total Connections

- [Metrics](#)
- [Troubleshooting](#)

Every connection to SQL Server spawns one or more processes, each uniquely identified by what SQL Server calls a SPID. The Total Connections display shows the number of user and system processes currently reported in the SQL Server instance. This number includes both active and inactive processes.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

You should view the total number of connections in light of the maximum number of processes allowed to connect to SQL Server. The user connections parameter specifies the maximum number of user processes that can simultaneously connect to a SQL Server instance.

Troubleshooting

If the total number of connections approaches the number of user connections limit then:

- 1 Edit the configuration file for SQL Server.
- 2 Increase the amount of user connections to a higher value.
- 3 Cycle SQL Server when possible to allow the new value to take effect.

Active Connections

The Active Connections statistic represents the total number of active and open threads reported on the server. This number displays the number of processes actively performing work.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

None.

Transactions

- [Metrics](#)

This statistic represents the total number of active transactions in the SQL Server instance that have not yet been committed, or are waiting on a blocking lock to complete.

TIP: Click this statistic to open the [Users performance category view](#).

NOTE: Performance Center displays this statistic on the Users performance category view as Active Transactions.

Metrics

None.

Current Locks

This statistic represents the total number of granted, converting, and waiting lock requests.

TIP: Click this statistic to drill down to the [All Lock tab](#) of the Locks view.

Network Vital Signs

The following network statistics are on the SQL Server Home view:

- [Logins](#)
- [Logouts](#)
- [Packets Received](#)
- [Packets Sent](#)

TIP: Click any of these statistics to open the [Network performance category view](#).

Packets Received

- [Metrics](#)

SQL Server counts incoming packets from the time the server is initiated. This statistic displays the delta count of all incoming packets received over the network since the last Performance Center sampling.

Metrics

None.

Packets Sent

- [Metrics](#)

SQL Server counts outgoing packets from the time the server is initiated. This statistic displays the delta count of all outbound packets written to the network the last Performance Center sampling.

Metrics

None.

Logins

- [Metrics](#)

This statistic represents the total number of logins started per second.

Metrics

None.

Logouts

- [Metrics](#)

This statistic represents the total number of logout operations started per second.

Metrics

None.

Memory Page Statistics

The Memory view includes the following statistics:

- [Ad Hoc SQL Hit Ratio](#)
- [Buffer Cache](#)
- [Buffer Cache Hit Ratio](#)
- [Buffer Cache Usage](#)
- [Memory Usage](#)
- [Procedure Hit Ratio](#)
- [Session Leaders - Memory](#)
- [SQL Cache](#)
- [SQL Cache Usage](#)

Related Topics

[Memory Detail View](#)

[Home View Statistics](#)

[Database Page Statistics](#)

[Contention Statistics](#)

[I/O Page Statistics](#)

[Network Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

[Users Page Statistics](#)

Buffer Cache Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

Data read from memory will produce end-user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the purposes of the SQL Server buffer/data cache.

The buffer cache hit ratio is a terrific indicator of how often user requests for data are satisfied through memory vs. being physically read from disk.

TIP: Click this statistic to drill down to the [Buffer Cache tab](#) of the Memory Detail view.

Metrics

To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, you should be aware that every server has its own 'personality' and might exhibit excellent performance with below average readings for the cache hit ratio. You should also be aware that excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

Consistently viewed low readings (60% or lower) might require tuning attention on the part of the DBA.

Troubleshooting

Ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server memory configuration option. Also, consider increasing your SQL Server Min. Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Often a user process is taking a large amount of memory due to an inordinate amount of I/O.

Procedure Plan Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

The SQL Server procedure cache is used to hold the execution plans for all Transact-SQL statements currently executing in the server. When a user executes a Transact-SQL statement, SQL Server looks in the procedure cache for a query plan to use.

This statistic is the percentage of query plan requests generated by stored procedures that are found in the procedure cache area. The percentage of times that a statement's plan and definition can be referenced in memory, the better the procedure execution time.

TIP: Click this statistic to drill down to the [SQL Cache tab](#) of the Memory Detail view.

Metrics

A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for code reference. Note that when a database is first started, the procedure cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

Troubleshooting

First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Ad Hoc SQL Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

When an ad hoc SQL statement is issued, the query plan is then stored in the SQL Server procedure cache area. If the identical ad hoc statement is launched in the future, SQL Server uses the query plan already stored in the procedure cache if it is still there. This statistic defines the percentage of times that a query plan for an ad hoc SQL statement is found in the procedure cache.

TIP: Click this statistic to drill down to the [SQL Cache tab](#) of the Memory Detail view.

Metrics

A high ad hoc hit rate is desirable, but is harder to maintain at a high level than something like a procedure cache hit rate. Therefore, an 80% or greater ad hoc cache hit rate is a good performance benchmark for code reference. Note that when a database is first started, the ad hoc cache hit rate will not be at an optimal level because all code being used will be relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the ad hoc cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

Troubleshooting

First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (To obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

SQL Cache

- [Metrics](#)

The total percentage of memory that SQL Server is using in its SQL/procedure cache. Microsoft has begun to transition from the term “procedure cache” to “SQL cache” to define this area of memory. The reason being that in SQL Server’s past, this area was devoted exclusively to holding query plans for stored procedures only.

The SQL Cache (procedure cache) is the part of the SQL Server memory pool that is used to store execution plans for Transact-SQL batches, stored procedures, and triggers. Execution plans record the steps that SQL Server must take to produce the results specified by the Transact-SQL statements contained in the batches, stored procedures, or triggers.

Metrics

None.

Memory Usage

- [Metrics](#)
- [Troubleshooting](#)

SQL Server is allocated memory at start up that will fall between the range of two configurable parameters: Min Memory and Max Memory. This statistic shows you the amount of SQL Server memory allocated at present. In a normal SQL Server system, this number fluctuates as SQL Server uses more memory and releases unused memory back to the operating system. SQL Server also works in conjunction with the Windows memory manager, deallocating space when it detects that the operating system is unable to satisfy memory requests of other operating system processes.

Metrics

An excessively high memory allocation may mean that the SQL Server instance is working in a memory deficient environment.

Troubleshooting

Increase the Max Memory parameter to provide SQL Server with more memory resources. Note that this may require adding physical RAM to the machine to satisfy the requests of both SQL Server and the Windows operating system.

Buffer Cache

- [Metrics](#)
- [Troubleshooting](#)

The total amount of memory, in megabytes, that is allocated for the SQL Server buffer cache. Each instance of SQL Server has its own buffer cache where it stores recently used data pages to reduce physical I/O. The goal is to make the buffer cache large enough to maximize the ratio of logical reads to physical reads, but not so large that excessive memory swapping starts generating physical I/O to the pagefile. (Instances of SQL Server 2000 do this automatically under the default configuration settings.)

TIP: Click either of these statistics to drill down to the [Buffer Cache tab](#) of the Memory Detail view.

Metrics

A percentage used consistently remaining close to 100% indicates a deficient amount of memory available to SQL Server.

Troubleshooting

First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

Session Leaders - Memory

- [Metrics](#)
- [Troubleshooting](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Frequently, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system level.

The leading memory session's display identifies the top users in the server with respect to memory consumption.

Metrics

If your SQL Server machine does not have an overabundance of memory, you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing.

Troubleshooting

You can use the Session Leaders - Memory statistic to find the users with the greatest current allocations of overall memory. Runaway processes can be immediately terminated from within the Performance Center Client.

Cache Usage

- [Metrics](#)
- [Troubleshooting](#)

The total MB of space that SQL Server is using in the buffer cache. Each instance of SQL Server has its own buffer cache where it stores recently used data pages to reduce physical I/O. The goal is to make the buffer cache large enough to maximize the ratio of logical reads to physical reads, but not so large that excessive memory swapping starts generating physical I/O to the pagefile. (Instances of SQL Server 2000 do this automatically under the default configuration settings.)

Metrics

A percentage used consistently remaining close to 100% indicates a deficient amount of memory available to SQL Server.

Troubleshooting

First, ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, an option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

SQL Cache Size

- [Metrics](#)

The total MB of space that SQL Server is using in its SQL/procedure cache. Microsoft has begun to transition from the term “procedure cache” to “SQL cache” to define this area of memory. The reason being that in SQL Server's past, this area was devoted exclusively to holding query plans for stored procedures only.

The SQL Cache (procedure cache) is the part of the SQL Server memory pool that is used to store execution plans for Transact-SQL batches, stored procedures, and triggers. Execution plans record the steps that SQL Server must take to produce the results specified by the Transact-SQL statements contained in the batches, stored procedures, or triggers.

Metrics

None.

SQL Cache Free

- [Metrics](#)

The total MB of space that is free in the SQL Server SQL/procedure cache. Microsoft has begun transitioning from the term “procedure cache” to “SQL cache” to define this area of memory. In the past this area was devoted exclusively to holding query plans for stored procedures only, but this is no longer the case.

The SQL Cache (procedure cache) is responsible for storing execution plans for Transact-SQL batches, stored procedures, and triggers. Execution plans record the steps that SQL Server must take to produce the results specified by the Transact-SQL statements contained in the batches, stored procedures, or triggers.

Metrics

None.

Memory Detail View

The Memory Detail view includes the following tabbed pages:

- [Buffer Cache](#)
- [Log Cache](#)
- [SQL Cache](#)
- [Leading Sessions](#)

Buffer Cache Tab

- [Metrics](#)

The buffer cache is a memory pool of buffer pages into which SQL Server reads data pages. The Buffer Cache tab of the Memory Detail view displays the top twenty objects that reside in the buffer cache. The table below describes the information available on the Buffer Cache tab of the Memory Detail view:

Information	Description
Database	The database name of where the object resides.
Owner	The owner of the object.
Table Name	The name of the table whose data pages have been read into the buffer cache.
Index Name	The name of the index whose pages have been read into the buffer cache.
Size (KB)	The size of the table or index in kilobytes.
Pinned	Whether or not the table has been pinned in memory.

Metrics

Keep a close eye on tables that have been pinned in memory. While it is wise to pin frequently-used, small lookup tables in memory, pinning larger tables can lead to overcrowding in the buffer cache and might result in thrashing behavior as SQL Server constantly moves pages in and out of the cache to satisfy user requests for data.

SQL Cache Tab

- [Metrics](#)

Beginning with SQL Server version 7.0, other SQL-related code objects might be placed in memory for reuse. Once SQL Server parses through and places a set of SQL or program code in memory, response time can be increased for subsequent calls to the same set of SQL or SQL code object. The SQL Cache tab of the Memory Detail view uses two grids to communicate the contents of the SQL/Procedure cache. The [first grid](#) displays the summary information and the [second grid](#) displays detail information about the SQL/Procedure cache. The table below describes the information available in the SQL Cache Summary grid on the SQL Cache tab of the Memory Detail view:

Column	Description
Cache Type	The category of SQL code inside the cache (ad hoc SQL, executable plan, etc.)
Object Count	The total objects for a particular cache type.
Cache Used (KB)	The total amount of the SQL/Procedure cache used by the particular cache type.

The table below describes the information available in the SQL Cache Details grid on the SQL Cache tab of the Memory Detail view:

Information	Description
Object Type	The type of object (ad hoc SQL, etc.)
Cache Type	The category of SQL code inside the cache (executable plan, etc.)
Name	The name of the object (if applicable).
Database	The database where the code originated.
User	The owner of the object or code.
Use Count	The number of uses for the object.
SQL Bytes	The amount of SQL bytes used by the code object.
Size (KB)	The size used by the object in kilobytes.
SQL	The actual SQL statement or code being executed.

Metrics

Seeing many objects of the ad hoc SQL type might indicate an environment where sessions are submitting a lot of SQL requests. Many SQL Server DBAs like to control code submissions through stored procedures.

Log Cache Tab

- [Metrics](#)
- [Troubleshooting](#)

Before ever writing transactions to disk, the log manager of SQL Server formats everything in memory. This area of memory is known as the log cache. The log writer of SQL Server moves through the log caches when transactions are committed (as well as other events) and flushes each cache out to disk. SQL Server also reads from the cache when log records are needed. The table below describes the information available on the Log Cache tab of the Memory Detail view:

Information	Description
Database	The name of the SQL Server database.
Log Cache Reads	The number of reads from the log cache.
Log Flushes	The number of times data was flushed from the log to disk.
Log Flush Waits	The number of times the log had to wait before flushing data to disk.
Log Flush Wait Time	The amount of time the log waited before flushing data to disk in milliseconds.
Log Growths	The number of times the log had to physically grow in size to meet the need for more space.
Log Shrinks	The number of times the log contracted in physical size.

Metrics

Seeing high amounts of wait time for log flushes could indicate a bottleneck at the disk level. A log that shows high number of growths likely indicates an undersized log. While automatic growth can alleviate out-of-space conditions, many growth operations can slow down overall operations. It is better to have a properly sized transaction log that allows SQL Server to continually enlarge it in size when needed.

Troubleshooting

Consider relocating logs showing high amounts of wait time to faster disks. For logs showing high numbers of growths, permanently enlarge the log(s) via an ALTER DATABASE command that will resize the log files.

Leading Sessions Tab

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. User connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system levels.

Performance Center displays information to help you find processes that are using the most memory on the server on the Leading Sessions tab of the Memory Detail view and the Memory tab of the Top Sessions view.

The table below describes the information available on these tabs:

Column	Description
PID	The process ID of the connected session.
User Name	The user name assigned to the process.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The machine name that originated the process.
Program	The program the process has invoked against SQL Server.
Memory Usage	The total number of memory pages allocated to the process.
Pct Mem Used	The percentage of overall memory among all processes that can be attributed to the process.
Database	The database in which the process is currently running.
Command	The command the process is currently issuing.
Open Trans	The number of open transactions for the process.
Blocked	The PID of any process blocking the current process.
Wait Time	The current amount of wait time for the process, in milliseconds.

NOTE: This information is available on both the Leading Sessions tab of the Memory Detail view and the [Memory tab](#) of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If your database server does not have an overabundance of memory, you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing.

I/O Page Statistics

The I/O view includes the following sections and statistics:

- [Checkpoint Pages](#)
- [Forwarded Record Fetches](#)
- [Full Scans](#)
- [I/O Error Rate](#)
- [Index Searches](#)
- [Log Cache Reads](#)
- [Log Flushes](#)
- [Page Reads](#)
- [Page Writes](#)
- [Probe Scans](#)
- [Range Scans](#)
- [Read Ahead Pages](#)
- [Server I/O Busy Rate](#)
- [Session Leaders - I/O](#)
- [Worktables Created](#)

Related Topics

[I/O Detail View](#)

[Home View Statistics](#)

[Database Page Statistics](#)

[Contention Statistics](#)

[Memory Page Statistics](#)

[Network Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

[Users Page Statistics](#)

Page Reads

- [Metrics](#)
- [Troubleshooting](#)

The Page Reads statistic represents that number of physical database page reads that are issued per second by SQL Server.

TIP: Click this statistic to drill down to the [Systems I/O tab](#) of the I/O Detail view.

Metrics

Page reads are to be expected, especially after initial server start up. This is because SQL Server must first satisfy requests for data and meta-data by reading information in from physical disk. Numerous page reads can also be expected if the physical server does not contain an adequate amount of memory to hold repetitively requested blocks of information.

No hard-and-fast rules exist for how many page reads per second is too much. You can cross-reference this statistic with the physical server disk statistics to see if physical page reads and accompanying physical disk I/O is approaching the server's premium capacity levels. And because logical I/O is always many times faster than physical I/O, you should also evaluate the buffer cache hit ratio to determine overall memory vs. physical read efficiency.

Troubleshooting

If you find that the server is becoming overworked from a physical I/O standpoint, there are several courses of action you can take:

- Examine index usage to ensure that unnecessary table scans are not occurring.
- Check the physical database design to see if table objects have been over-normalized.
- Ensure that SQL Server is configured to use sufficient amounts of memory. Examine the min server memory (MB) and max server memory (MB) parameters to see if SQL Server is constrained on either end of the memory spectrum.
- Check for large pinned table objects that could be using excessive amounts of space in the buffer cache.
- Last, but not least, investigate the possibility of adding more RAM to the physical server.

Page Writes

- [Metrics](#)

The Page Writes statistic represents the number of physical database page writes issued by SQL Server. Page Writes take place during operations such as checkpoints, lazywriter writes, index creations, and BCP routines.

TIP: Click this statistic to drill down to the [Systems I/O tab](#) of the I/O Detail view.

Metrics

Page Writes can give you an idea of overall physical write activity, however there are a number of statistics that pertain specifically to certain write activities like checkpoints, etc., that you can examine to determine the amount of physical writes caused by distinct SQL Server processes.

With respect to performance, response times experienced by SQL Server users are normally not impacted by write operations unless the writes are synchronous in nature. These are typically BCPs, database recovery operations, and index creations.

Read Ahead Pages

- [Metrics](#)

The Read Ahead Pages statistic represents the number of physical database page read in anticipation of use by SQL Server. If SQL Server senses that database pages are being read in a sequential manner, it will institute a pre-fetch mechanism that moves pages into the buffer cache before they are actually needed by a process.

TIP: Click this statistic to drill down to the [Systems I/O tab](#) of the I/O Detail view.

Metrics

If data is accessed sequentially (for example, through the use of a clustered index), the read ahead mechanism of SQL Server can increase performance by needed database pages already in the buffer cache before they are actually requested.

However, because the read ahead mechanism is typically triggered by full table or index range scans, if the read ahead pages are actually required to satisfy a user's query, then performance might actually suffer. In these cases, the judicious use of indexes is a better route to take.

Log Flushes

- [Metrics](#)
- [Troubleshooting](#)

The Log Flushes statistic represents the total number of log pages for all databases written to disk by the log writer process. A log flush occurs when SQL Server writes all changes from the database's log cache out to the database's log files on disk.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

Increasing numbers observed for log flushes should not cause concern unless the I/O subsystem of the server appears overwhelmed. In addition, to minimize I/O contention between a database and its accompanying log, it is wise to place database files and log files on separate disks.

Troubleshooting

If you have placed a very active database on the same physical file as its log, you can look into moving the log to a separate physical device by adding new log files to a new drive and subsequently removing the old log files when they are not being used.

Log Cache Reads

- [Metrics](#)

The Log Cache Reads statistic represents the reads performed per second through the log manager cache. Before ever writing transactions to disk, the log manager of SQL Server formats them in memory. This area of memory is known as the log cache and only contains log records for SQL Server 2000 and later. The log writer of SQL Server moves through the log caches when transactions are committed (as well as other events) and flushes each cache out to disk.

Metrics

None.

Checkpoint Pages

- [Metrics](#)
- [Troubleshooting](#)

The Checkpoint Pages statistic represents the number of pages flushed to disk per second by a checkpoint or other operation that require all dirty (modified) pages to be flushed.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

Checkpoint operations are used by SQL Server to minimize the amount of work the server must perform when databases are recovered during system startup. Checkpoints periodically write out modified pages that are found in the buffer cache to disk. Afterwards, SQL Server records the operation in the log to signify that the operation succeeded.

Checkpoints can be explicitly performed by a database owner issuing the checkpoint command. SQL Server also performs checkpoints automatically for databases that have the trunc log on chkpt option set.

Large SQL Servers have the potential to generate lots of checkpoint write operations. Although SQL Server will do what it can to minimize checkpoint activity, you can also set the recovery interval server parameter to influence how often checkpoints should run.

Troubleshooting

If you believe excessive checkpoint activity is occurring, you can take the following steps:

- Set the recovery interval server parameter to a larger value with sp_configure.
- Restart SQL Server so the change will take affect or use the RECONFIGURE option to make the change immediately.

Full Scans

- [Metrics](#)
- [Troubleshooting](#)

The Full Scans statistic represents the total number of full table or index scans per second.

Metrics

Full scans occur if a table is inadequately indexed or if SQL Server truly needs to access all rows in a table or index to satisfy a query. Other operations that can cause full scans include UPDATE STATISTICS calls.

Unnecessary scans on large tables is something to avoid, and can be a signal to you as a DBA to investigate the use of more indexes and to review SQL access through EXPLAIN plans. Small table scans are actually a good thing because SQL Server can often cache the entire table in a single I/O operation. Large numbers of index scans are normally desirable too, because it typically indicates the fastest possible resolution to data access requests.

Troubleshooting

Here are some methods you can use to avoid unnecessary large table scans:

- Try not to use SQL statements that include the NOT IN, NOT LIKE, <>, IS NULL operators because they typically suppress the use of indexes.
- When referencing concatenated indexes with queries, be sure the leading column in the index is used. If it is not, the index will not be used at all.
- Avoid using functions in WHERE predicates.

Range Scans

- [Metrics](#)

The Range Scans statistic represents the total number of qualified range scans through indexes per second.

Metrics

Large numbers of index scans are normally desirable, because it typically indicates the fastest possible resolution to data access requests is being taken.

Index Searches

- [Metrics](#)

The Index Searches statistic represents the total number of index searches per second. Index searches are normally used to start range scans, for single index record fetches and can be used to reposition an index.

TIP: Click this statistic to drill down to the [Systems I/O tab](#) of the I/O Detail view.

Metrics

Large numbers of index searches are normally desirable because they typically indicate the fastest possible resolution to data access requests is being taken.

Probe Scans

- [Metrics](#)

The Probe Scans statistic represents the total number of probe scans per second. Probe scans are used in SQL Server to directly find rows in an index or base table.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

Large numbers of probe scans are normally desirable, because they typically indicate the fastest possible resolution to data access requests is being taken.

Worktables Created

- [Metrics](#)

The Worktables Created statistic represents the total number of work tables created per second. Worktables are used many times by SQL Server to perform a logical operation specified in an end user SQL statement. GROUP BY, ORDER BY, or UNION queries can cause worktables to be created as can specific CREATE statements used in Transact SQL processing.

Worktables are built in the tempdb database and are dropped automatically at the end of the statement or procedure run.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

The only thing to keep in mind with respect to worktables is that the tempdb database should be large enough to hold large worktables.

Forwarded Record Fetches

- [Metrics](#)
- [Troubleshooting](#)

The Forwarded Record Fetches statistic represents the total number of records per second fetched by reading forwarded record pointers.

SQL Server moves rows in a table under certain conditions. One situation might arise when you update a row in a table that has a variable-length column to a larger size that no longer fits on its original page. Another situation would be if SQL Server moves a row when the clustered index column changes.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

When SQL Server creates a forwarding pointer, it remains in place unless one of two things happens. The first is when a row shrinks enough to move back to its original location. The second is when the entire database shrinks. When a database file shrinks, SQL Server reassigns the row identifiers, which are used as the row locators, so the shrink process never generates forwarded rows.

Forwarded records can reduce performance at times because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself is read.

Troubleshooting

If consistent numbers are present for Forward Record Fetches, examine your databases to see which tables have forwarded records.

To see the total count of forwarded records in a table, you can enable trace flag 2509, and then execute the DBCC CHECKTABLE command. The output should display the number of forwarded records in that table. Tables with many forwarded records could be candidates for table reorganization.

Server I/O Busy Rate

- [Metrics](#)

The Server I/O Busy or I/O Busy statistic represents the time in milliseconds that the database server has spent performing input and output operations since the last refresh in Performance Center.

TIP: Click this statistic to drill down to the [Systems I/O tab](#) of the I/O Detail view.

Metrics

None.

Session Leaders - I/O

- [Metrics](#)

Heavy I/O activity in a system can indicate that the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are responsible for 75% or more of the I/O activity. Your system may be running a large batch load or other typical processes, which are perfectly okay. On the other hand, it may be a runaway process or other rogue connection, which you need to track down and possibly eliminate.

The leading I/O session's display lets you see who the leading sessions are in your system with respect to I/O.

NOTE: This statistic displays on both the [Users performance category view](#) and the I/O performance category view.

TIP: Click this statistic to drill down to the [Systems I/O tab](#) of the I/O Detail view.

TIP: Click an item in the chart to open the [Overview tab](#) of the Session Detail view.

Metrics

One or two users consuming more than 75% of the total I/O load can indicate a runaway or improper process. You can drill into the I/O activity of all users and quickly see if this is the case.

Total I/O

The Total I/O statistic represents the total number of physical reads and writes.

Metrics

None.

I/O Error Rate

- [Metrics](#)
- [Troubleshooting](#)

I/O Error Rate reflects total number of I/O errors (errors during read and write operations) encountered by the server since the last refresh inside Performance Center. I/O Error Rate is a percentage based on Total I/O (the sum the physical reads and writes).

TIP: Click this statistic to drill down to the [User I/O tab](#) of the I/O Detail view.

Metrics

You should observe few, if any errors.

Troubleshooting

If you notice any errors, you should check the SQL Server error log for details.

I/O Detail View

The I/O Detail view includes the following tabbed pages:

- [Database I/O](#)
- [File I/O](#)

- [Leading Sessions](#)
- [System I/O](#)
- [User I/O](#)

System I/O Tab

- [Metrics](#)
- [Troubleshooting](#)

SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The System I/O tab of the I/O Detail view details space-related I/O operations. The table below describes the information available in the System I/O tab:

Information	Description
Pages/Extents Allocated	The number of space extents that SQL Server allocated. Rapidly increasing numbers for these statistics indicates that SQL Server is receiving large volumes of incoming data and is allocating space to make room.
Page/Extent Deallocations	This indicates that SQL Server is reclaiming space from database objects due to shrinking database volumes.
Freespace Scans	The number of scans performed by SQL Server to locate free space for an incoming record.
Page Splits	When data is inserted or updated in a table, SQL Server might reorganize the storage of the data in the table's index pages. When an index page becomes full, but a DML operation demands room on that page, SQL Server moves about half the rows to a new page to accommodate the request. This reorganization is known as a page split. Performance for DML actions can be impaired from page split operations. In addition, more index pages can make for longer index scan times.

Metrics

Increasing numbers for extent and page allocation, and freespace operations likely indicates aggressive volumes of data being inserted or modified in SQL Server.

Page splits cause additional overhead in the form of CPU usage and I/O. Observing large numbers of page splits can signal a resource bottleneck in your server.

Troubleshooting

To avoid page splits, you can look into tuning the FILLFACTOR property of an index, which controls the percentage of the index page that is filled during creation. The default, 100, tells SQL Server to completely fill each page, whereas lower numbers tell SQL Server to leave room for additional index rows or updates to existing rows.

Database I/O Tab

SQL Server performs many system-related I/O functions to keep data moving into and out of the server. The Database I/O tab of the I/O Detail view details maintenance-related I/O operations. The table below describes the information available in the Database I/O tab

Information	Description
Database	The name of the database.
DBCC Logical Scans	The number of logical read scan bytes per second caused by DBCC operations.

Information	Description
Bulk Copy Rows	The number of rows copied either into or out of the database via the BCP utility.
Bulk Copy Throughput	The amount of data (in KB) copied via BCP operations.
Backup/Restore Throughput	The read/write throughput for backup and restore operations of a database.
Log Cache Reads	The number of reads performed through the log manager cache.
Log Flushes	The number of log flushes for the server.
Backup/Restore T-Put	Defines the read/write throughput for backup and restore operations.
Log Growths	The number of times the transaction log for the database has been expanded.
Log Shrinks	The number of times the transaction log for the database has been reduced in size.
Log Truncations	The number of times the transaction log for the database has been truncated (possibly by log backup operations).

User I/O Tab

- [Metrics](#)
- [Troubleshooting](#)

The User I/O tab of the I/O Detail view displays statistics that track various user-related I/O operations. The I/O function along with its counter value is presented. The User I/O tab details performance statistics that reflect how SQL Server is performing object access operations. The table below describes the information available on the User I/O tab of the I/O Detail view:

Information	Description
Forwarded Records	The number of records per second fetched through forwarded record pointers. At times forwarded records can reduce performance because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself is read.
Full Scans	Full scans of moderately sized indexes or tables are generally okay. SQL Server can scan and cache a small table much faster than using its index to navigate to any requested data. Full, unrestricted, large table scans, however, are typically not good and degrade overall system performance and response time.
Index Searches	The total number of index searches per second. Index searches are normally used to start range scans, for single index record fetches and can be used to reposition an index.
Probe Scans	The total number of probe scans per second. Probe scans are used in SQL Server to directly find rows in an index or base table.
Range Scans	The total number of qualified range scans through indexes per second.
Skipped Ghosted Records	The number of ghosted records per second skipped during scans.

Metrics

Full scans occur if a table is inadequately indexed or if SQL Server truly needs to access all rows in a table or index to satisfy a query. UPDATE STATISTICS calls can also cause full scans.

Unnecessary scans on large tables is something to avoid, and can be a signal to you as a DBA to investigate the use of more indexes and to review SQL access through EXPLAIN plans. Small table scans are actually a good thing because SQL Server can often cache the entire table in a single I/O operation. Large numbers of index scans are normally desirable too, because this typically indicates the fastest possible resolution to data access requests.

When SQL Server creates a forwarding pointer, it remains in place unless one of two things happens. The first is when a row shrinks enough to move back to its original location. The second is when the entire database shrinks. When a database file shrinks, SQL Server reassigns the row identifiers, which are used as the row locators, so the shrink process never generates forwarded rows. Forwarded records can reduce performance at times because additional I/O is involved to first obtain the record pointer to the relocated row, and then the row itself is read.

Large numbers of index searches and probe scans are normally desirable because they typically indicate the fastest possible resolution to data access requests is being taken.

Troubleshooting

Here are some methods you can use to avoid unnecessary large table scans:

- Try not to use SQL statements that include the NOT IN, NOT LIKE, <>, IS NULL operators because they typically suppress the use of indexes.
- When referencing concatenated indexes with queries, be sure the leading column in the index is used. If it is not, the index will not be used at all.
- Avoid using functions in WHERE predicates.

If consistent numbers are present for Forward Record Fetches, examine your databases to see which tables have forwarded records. This can easily be done with the Embarcadero Space Analyst component of DBArtisan. If you do not have Space Analyst, then to see the total count of forwarded records in a table, enable trace flag 2509, and then execute the DBCC CHECKTABLE command. The output should display the number of forwarded records in that table. Tables with many forwarded records could be candidates for table reorganization.

File I/O Tab

- [Metrics](#)

The File I/O tab of the I/O Detail view summarizes I/O activity for each database file, letting you quickly spot the “hot” databases and files on your server. The table below describes the information available on the File I/O tab of the I/O Detail view for SQL Server 2000 and later:

Information	Description
Database	The database name.
File ID	The file identifier for the target file.
Logical Name	The name given the file by the DBA.
File Name	The physical file name of the file.
Timestamp	The internal time stamp of when the data was obtained.
Reads	The number of reads issued against the database file.
Writes	The number of writes issued against the database file.
Bytes Read	The total number of bytes read for the database file.
Bytes Written	The total number of bytes written for the database file.
I/O Stall	The total amount of time that processes have waited for I/O operations to complete, in milliseconds.

NOTE: Data is only available for SQL Server 2000 and later.

Metrics

Consider moving databases and/or files with lots of I/O activity and wait time onto separate drives/devices.

Leading Sessions Tab

- [Metrics](#)

When a system undergoes heavy I/O activity, sometimes you find that all the user connections are contributing somewhat equally to the overall load. Frequently, however, one or two user connections are responsible for 75% or more of the I/O activity. It may be that a large batch load or other typical process is running that is perfectly okay for your system. Alternatively, it may be a runaway process or other rogue connection that you should track down and possibly eliminate.

Performance Center displays information to help you find processes that are using the most memory on the server on the Leading Sessions tab of the I/O Detail view and the I/O tab of the Top Sessions view.

The table below describes the information available on these tabs:

Column	Description
PID	The process ID.
User Name	The user name assigned to the process.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Physical I/O	The current cumulative number of reads and writes issued by the process.
Pct I/O Used	The percentage of overall I/O that can be attributed to the process.
Database	The database where the process is running.
Command	The command the process is currently issuing.
Open Trans	The number of open transactions for the process.
Blocked	The PID of any process blocking the current process.
Wait Time	The current amount of wait time for the process, in milliseconds.

NOTE: This information is available on both the Leading Sessions tab of the I/O Detail view and the [I/O tab](#) of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Finding one more two users that are consuming more than 75% of the total I/O load can indicate a runaway or improper process. By drilling down into the I/O activity of all users, you can quickly see if this is the case.

Space Page Statistics

The Space view includes the following statistics:

- [Database Overview](#)
- [Log Overview](#)
- [Disk Free Space](#)

Related Topics[Space Detail View](#)[Home View Statistics](#)[Database Page Statistics](#)[Contention Statistics](#)[I/O Page Statistics](#)[Memory Page Statistics](#)[Network Statistics](#)[OS Page Statistics](#)[Top SQL Page Statistics](#)[Users Page Statistics](#)

Disk Free Space

- [Metrics](#)
- [Troubleshooting](#)

The Disk Free Space statistic provides a summary of the total database and log space used per server disk drive, as well as a summary of free disk space for each server drive.

Metrics

If any database or transaction log file has been set up to automatically grow, the DBA should ensure there is enough server disk space to accommodate any new, additional requests for space.

Troubleshooting

If you see any drive that has reached zero free space (or is close), you might want to add new files, on other disks with abundant free space, to any databases or transaction logs so that no out-of-space errors result.

Database Overview

- [Metrics](#)
- [Troubleshooting](#)

A SQL server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The Databases Low on Space statistic represents a count of databases that have fallen below the Performance Center recommended free space limit.

TIP: Click this category heading to drill down to the [Fragmentation tab](#) of the Space Detail view.

Metrics

If a database's free space goes below the Performance Center recommended threshold, (and either the database or transaction log does not have their automatic growth property enabled or the files have reached their growth limit) then the DBA should take action to ensure that the database or transaction log does not run out of available free space.

Troubleshooting

If the percent used amount of a database is approaching problematic levels, then there are three ways a DBA can rectify the situation:

- 1 The DBA can resize the current file(s) used by the database via an ALTER DATABASE ... MODIFY FILE command.
- 2 The DBA can add a new file to the database via the ALTER DATABASE ... ADD FILE command.
- 3 The DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE ... MODIFY FILE ... FILEGROWTH command. You should also ensure that the MAXSIZE setting of each file is set appropriately.

Of course, the DBA should also ensure that enough physical space exists on the server to accommodate additional database space.

Log Overview

- [Metrics](#)
- [Troubleshooting](#)

A SQL server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The Log Overview statistic represents a count of database transaction logs that have fallen below the Performance Center recommended free space limit.

Metrics

If a database's transaction log free space goes below the Performance Center recommended threshold, (and the transaction log does not have their automatic growth property enabled or the files have reached their growth limit) then the DBA should take action to ensure that the transaction log does not run out of available free space.

Troubleshooting

There are several things a DBA can do to ensure that a database's log does not run out of available free space:

- 1 First, most transactional-oriented databases should have their logs assigned to a separate physical drive than the database. Reasons for doing so include:
 - It prevents competition for space between the log and the database itself.
 - It allows the log to be monitored for space more effectively.
 - It improves performance.
- 2 If the database is not critical in nature, you can set the truncate log on checkpoint option (trunc log on chkpt), which will eliminate any non-active space in the log when a database checkpoint occurs.
- 3 Critical databases needing higher levels of recovery should have schedules established that regular perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.
- 4 If a critical transaction log becomes full, it might be impossible to use standard procedures to dump transactions and reclaim space. The dump operation will likely have to incorporate the no log or truncate only options.
- 5 If a transaction log continuously approaches dangerously low levels of free space, then the DBA should allow the underlying file(s) of the log to automatically grow to meet the demand. This can be accomplished by using the ALTER DATABASE ... MODIFY FILE ... FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

The DBA should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log since only non-active space in the transaction log is removed from log dumps or truncation operations.

Space Detail View

The Space Detail view includes the following tabbed pages:

- [Disk Space](#)
- [Indexes](#)
- [File Groups/Files](#)
- [Fragmentation](#)
- [Tables](#)
- [Virtual Log Files](#)

File Groups/Files Tab

- [Metrics](#)
- [Troubleshooting](#)

SQL Server manages physical storage space through files and file groups. The file groups tab displays detailed information regarding storage usage in each of the SQL Server file groups. The File Groups/Files tab of the Space Detail view displays space measures for the file groups and files of a particular database. The [first grid](#) presents file group information and the [second grid](#) displays data for the individual files in a database. The table below describes the information available in the File Group Summary grid on the File Group/Files tab of the Space Detail view:

Information	Description
File Group	The name of the file group.
Files	The number of files that make up the file group.
Size (MB)	The total physical size of the file group.
Table Reserved	The amount of reserved space consumed by tables.
Index Reserved	The amount of reserved space consumed by indexes.

The table below describes the information available in the Files Summary section of the Files tab in the File Summary grid on the File Group/Files tab of the Space Detail view:

Information	Description
Logical Name	The nickname given to the file by the DBA.
File Group	The name of the file group.
File Name	The physical name of the file.
File Name	The name and location of the file.
Size (MB)	The total physical size of the file.
Growth	This indicates if the file can automatically grow in size.
Growth Amount	This indicates how much the file will grow in size.

Information	Description
Max Size	This indicates the maximum file size that the file can grow.
Total Extents	The total number of extents taken up by the file.
Used Extents	The actual number of used extents in the file.

Metrics

Unless space is tight on a server, it is normally wise practice to allow your files to automatically grow to meet demand for more incoming data. It is also smart to physically separate your database and log files onto separate physical drives.

Troubleshooting

To let your files automatically grow until out of space, you can easily do this by setting the file's growth option in Embarcadero's DBArtisan or Embarcadero's Rapid SQL, or by using the ALTER DATABASE... MODIFY FILE command.

Virtual Log Files Tab

- [Metrics](#)

The Virtual Log Files tab of the Space Detail view displays an internal structure of sorts for a each database's log. The presented information is helpful when trying to shrink a database's log because you can see how much of the log is active and exactly where the active portion resides. The table below describes the information available on the Virtual Log Files tab of the Space Detail view:

Information	Description
Database Name	The name of database.
File Name	The name of the log file.
Status	This indicates if this portion of the log is active or inactive (not being used).
Size	The size of this portion of the log in MBs.

Metrics

None.

Tables Tab

- [Metrics](#)

Tables and indexes consume the storage in all databases. The Tables tab displays summary information regarding table and index storage for all databases.

The Tables tab of the Space Detail view displays the space details for tables in a selected database. The table below describes the information available on the Tables tab of the Space Detail view:

Column	Description
Owner	The owner of the table.
Table Name	The name of the table.
File Group	The file group where the table resides.

Column	Description
Rows	The number of rows in the table.
Reserved (KB)	The amount of space reserved for the table in kilobytes.
Used (KB)	The amount of space the table is using.
Free (KB)	The amount of free space that the table possesses.
% of Database	The percentage of the database that the table consumes.

Metrics

Negative numbers viewed for index space information can be caused by inaccuracies contained in the SQL Server data dictionary. Frequently, running a DBCC UPDATEUSAGE command against the database resolves the problem. However, there are bugs in SQL Server that sometimes caused erroneous numbers to be reported for the reserved space amounts used by tables and indexes.

Indexes Tab

- [Metrics](#)

Tables and indexes consume the storage in all databases. The Database Object Detail grid displays object space details for the database selected in the Database Object Summary grid.

The Indexes tab of the Space Detail view displays the space details for indexes in a selected database. The table below describes the information available on the Indexes tab of the Space Detail view:

Information	Description
Table Name	The owner/table that is the parent of the index.
Index Name	The name of the index.
Index Type	The type of index (clustered, etc.)
File Group	The file group where the object resides.
Reserved (KB)	The amount of space reserved by the object in kilobytes.
Used (KB)	The amount of space used by the object in kilobytes.
Free (KB)	The amount of free space used by the object in kilobytes.
% of Database	The percentage of the database that the index consumes.

Metrics

Negative numbers viewed for index space information can be caused by inaccuracies contained in the SQL Server data dictionary. Frequently, running a DBCC UPDATEUSAGE command against the database resolves the problem. However, there are bugs in SQL Server that sometimes caused erroneous numbers to be reported for the reserved space amounts used by tables and indexes.

Fragmentation Tab

- [Metrics](#)
- [Troubleshooting](#)

Object fragmentation can hinder the performance of SQL Server in a couple of ways. The first type of fragmentation is internal fragmentation. In a nutshell, this means that an index is taking up more space than it really needs. Larger indexes make for longer scan times, so performance degradation can result for indexes with high amounts of internal fragmentation. External fragmentation is the second form of fragmentation, and occurs when the logical order of pages for an index do not match the actual physical order. It also presents itself when the extents belonging to a table are not contiguous. External fragmentation generally only degrades performance when SQL Server is performing an ordered scan of a table or index.

The Fragmentation tab of the Space Detail view presents fragmentation information for the tables and indexes of a selected database. The table below describes the information available on the Fragmentation tab of the Space Detail view:

Column	Description
Table Name	The name of the table.
Index Name	The name of the index.
Object Type	The type of object (table or index).
Local Fragmentation	The percentage of extent fragmentation found after scanning the leaf pages of an index. Logical fragmentation is not relevant to tables or text indexes. Lesser values indicate little fragmentation.
Scan Density	Cannot be calculated adequately if the table resides on more than one file. However, if the table is placed on only one file, the Scan Density lists a percentage of the amount of external/extent fragmentation. A value of 100 represents a perfectly contiguous object. Values of less than 100 indicate that some fragmentation exists.
Average Page Density	In general, a low value indicates little internal fragmentation. Page Density represents how full the actual pages are.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

In general, you want to see low values for logical fragmentation and page density, and high values for scan density.

Troubleshooting

If you observe threatening fragmentation levels, you can take many courses of action:

- 1 You can use DBCC DBREINDEX to rebuild all the tables for a selected table or selectively rebuild just the indexes you want.
- 2 For SQL Server 2000 and later, you can use DBCC INDEXDEFRAG to rebuild the indexes for a table online. Users can perform DML operations on the objects involved in the DBCC operation.
- 3 You can drop and recreate the indexes that appear fragmented.

Disk Space Tab

- [Metrics](#)
- [Troubleshooting](#)

The Disk Space tab of the Space Detail view displays the amounts of space used by SQL Server per physical drive and by database per physical drive. The [first section](#) displays summary information and the [second section](#) contains space information per database.

The table below describes the information available in the Disk Summary by Space section on the Disk Space tab of the Space Detail view:

Column	Description
Disk Drive	The physical drive letter.
SQL/Data Space (MB)	The amount of reserved database space on the drive.
SQL/Log Space (MB)	The amount of reserved log space on the drive.
Disk Free Space (MB)	The total amount of free space that remains on the drive.

The table below describes the information available in the Disk Summary By Database section on the Disk Space tab of the Space Detail view:

Column	Description
Disk Drive	The physical drive letter.
Database	The database name.
SQL/Data Space (MB)	The amount of reserved database space on the drive.
SQL/Log Space (MB)	The amount of reserved log space on the drive.

Metrics

If you allow your database and/or log files to automatically grow, and you see little or no free space left on their physical drives, an option is to add new files to the database or log on different drives to avoid any out of space errors.

It is also smart to physically separate your database and log files onto separate physical drives.

Troubleshooting

If you need to add new files to your databases or logs, you can do so easily by using the ALTER DATABASE... ADD FILE and ALTER DATABASE... ADD LOG FILE commands.

Users Page Statistics

The Users view includes the following statistics:

- [Active Connections](#)
- [Active Transactions](#)
- [Auto-Param Attempts](#)
- [Failed Auto-Param Attempts](#)
- [Inactive Connections](#)
- [Session Leaders - Memory](#)
- [Session Leaders - I/O](#)
- [Session Leaders - CPU](#)
- [SQL Compilations](#)
- [SQL Re-Compilations](#)

- [System Connections](#)
- [T-SQL Batches](#)
- [Total Connections](#)
- [Users Blocked](#)

Related Topics

[Users Detail View](#)

[Home View Statistics](#)

[Database Page Statistics](#)

[Contention Statistics](#)

[I/O Page Statistics](#)

[Memory Page Statistics](#)

[Network Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

Total Connections

- [Metrics](#)
- [Troubleshooting](#)

Every connection to SQL Server spawns one or more processes, each uniquely identified by what SQL Server calls a SPID. The Total Connections display shows the number of user and system processes currently reported in the SQL Server instance. This number includes both active and inactive processes.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

You should view the total number of connections in light of the maximum number of processes allowed to connect to SQL Server. The user connections parameter specifies the maximum number of user processes that can simultaneously connect to a SQL Server instance.

Troubleshooting

If the total number of connections approaches the number of user connections limit then:

- 1 Edit the configuration file for SQL Server.
- 2 Increase the amount of user connections to a higher value.
- 3 Cycle SQL Server when possible to allow the new value to take effect.

Active Transactions

- [Metrics](#)

The active transactions statistic represents a count of the number of in-process transactions for SQL Server.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

None.

T-SQLBatches

- [Metrics](#)

The T-SQL batches statistic refers to the number of transact SQL batches processed by SQL Server.

Metrics

None.

Active Connections

- [Metrics](#)

The Active Connections statistic represents the total number of active and open threads reported on the server. This number displays the number of processes actively performing work.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

None.

Inactive Connections

- [Metrics](#)

The Inactive Connections statistic represents the total number of threads logged on to the server that are idle at the current time.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

None.

System Connections

- [Metrics](#)

The System Connections statistic represents the total number of threads logged on to the server that are SQL Server internal processes.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

None.

- [Troubleshooting](#)

SQL Compilations

- [Metrics](#)
- [Troubleshooting](#)

The number of SQL compilations performed per second, indicating the number of times the compile code path is entered. This also includes compiles due to recompiles. When SQL Server user activity levels become stable, this value reaches a steady state.

Metrics

Because compilation is a significant part of a query's turnaround time, you should strive to have as many compilations stored in the cache as possible. If this number does not stabilize in direct proportion to user activity stabilizing, you should investigate your SQL Cache to see if it has adequate memory assigned to it.

Troubleshooting

If you discover that the SQL Cache area is undersized, first ensure SQL Server is configured to use as much physical memory as possible by checking the Max Server Memory configuration option. Also, consider increasing your SQL Server Min Memory parameter to allocate more memory to SQL Server. (Note that to obtain optimal values for these parameters, one option is to install more physical RAM to your server.) Check for any large objects that are pinned in memory that could possibly be removed.

SQL Re-Compilations

- [Metrics](#)
- [Troubleshooting](#)

The SQL re-compilations statistic represents the total number of recompiles triggered per second in a SQL Server instance. Recompiles occur when SQL Server determines that the currently defined execution plan for an executing stored procedure might no longer be the best possible plan. SQL Server pauses the query execution and recompiles the stored procedure.

Metrics

Recompiles slow down the process that is executing the procedure and increase the load on the CPU. By extension, the more recompiles that are occurring on your system, the more overall load increases, resulting in poor performance. In general, you want to keep the number of recompiles low. The most common reasons SQL Server would issue a recompile are: Running `sp_recompile` against any table referenced in the stored procedure, significant data changes in a referenced table, schema changes to referenced objects, the use of the `WITH RECOMPILE` clause in the `CREATE PROCEDURE` or `EXECUTE` statement, and a plan no longer available in the system cache.

Troubleshooting

Try to practice coding standards that eliminate the most frequent causes detailed above. Also, try to:

- Use temporary tables only in the stored procedure that created them.
- Minimize creating temporary tables in control block structures.
- Use the KEEP PLAN option on references to static temporary tables.
- Issue the CREATE TABLE statement before any other references to the created table.
- Minimize the use of temporary tables.

Auto-Param Attempts

- [Metrics](#)
- [Troubleshooting](#)

Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. This statistic shows the number of auto-parameterization attempts per second and includes failed, safe, and unsafe auto-parameterizations.

Metrics

SQL Server's ability to match new SQL statements with existing, unused execution plans is increased when parameters or parameter markers are used in Transact-SQL statements. If an SQL statement is executed without parameters, SQL Server parameterizes the statement internally to increase the possibility of matching it against an existing execution plan. A high number for this statistic shows that SQL Server is efficiently reusing existing cached plans.

Troubleshooting

You can increase the ability of the relational engine to match complex SQL statements to existing, unused execution plans, by explicitly specify the parameters using either `sp_executesql` or parameter markers in your T-SQL code.

Failed Auto-Param Attempts

- [Metrics](#)
- [Troubleshooting](#)

Auto-parameterization occurs when an instance of SQL Server attempts to reuse a cached plan for a previously executed query that is similar to, but not the same as, the current query. The Failed Auto-Param Attempts statistic shows the number of failed auto-parameterization attempts per second.

Metrics

SQL Server's ability to match new SQL statements with existing, unused execution plans increases when parameters or parameter markers are used in Transact-SQL statements. If an SQL statement is executed without parameters, SQL Server parameterizes the statement internally to increase the possibility of matching it against an existing execution plan. A small number for this statistic shows that SQL Server is efficiently reusing existing cached plans.

Troubleshooting

You can increase the ability of the relational engine to match complex SQL statements to existing, unused execution plans, by explicitly specify the parameters using either `sp_executesql` or parameter markers in your T-SQL code. Doing so helps lower this number.

Users Blocked

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock will wait indefinitely.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

You should immediately investigate any indicator above zero, before the situation has a chance to mushroom. If this number is consistently greater than zero, investigate by using the Blocking Locks tab of the Locks view.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is where it gets tricky. The DBA can drill down into user detail and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process will wait for a lock before timing out.

Session Leaders - Memory

- [Metrics](#)
- [Troubleshooting](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Frequently, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system levels.

The leading memory session's display identifies the top users in the server with respect to memory consumption.

NOTE: This statistic displays on both the [Memory performance category view](#) and the Users performance category view.

Metrics

If your SQL Server machine does not have an overabundance of memory, you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing.

Troubleshooting

You can use the Session Leaders - Memory statistic to find the users with the greatest current allocations of overall memory. Runaway processes can be immediately terminated from within the Performance Center Client.

Session Leaders - I/O

- [Metrics](#)

When a system undergoes heavy I/O activity, sometimes you find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are responsible for 75% or more of the I/O activity. This may be because it may be that a large batch load or other typical process is running that is perfectly okay for your system. Or it may be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

The leading I/O session's display allows you to see who runs the leading sessions in your system with respect to I/O.

NOTE: This statistic displays on both the [I/O performance category view](#) and the Users performance category view.

TIP: Click this statistic to drill down to the [Leading Sessions tab](#) of the I/O Detail view.

Metrics

Finding one more two users that are consuming more than 75% of the total I/O load can indicate a runaway or improper process. By drilling down into the I/O activity of all users, you can quickly see if this is the case.

Session Leaders - CPU

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Often, user connections can get out of hand with CPU use, and extreme cases have caused headaches at both the database and operating system levels.

The leading CPU session's display allows you to see who runs the leading sessions in your system with respect to CPU usage.

Metrics

Finding one or two users who use the majority of the CPU can indicate runaway or improper processes. By drilling down into the CPU activity of all users, you can quickly see if this is the case.

Users Detail View

The Users view Detail includes the following tabbed pages:

- [All Locks](#)
- [Blocking Locks](#)
- [Processes](#)

Processes Tab

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Often, user connections can get out of hand with memory consumption, CPU, or physical I/O, and extreme cases have caused headaches at both the database and operating system levels.

The Processes tab of the Users Detail view displays information to help pinpoint problem processes. The table below describes the information available on the Processes tab of the Users Detail view:

Column	Description
SPID	The unique SQL Server process ID of the process holding the lock.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Login	The SQL Server login name of the process.
NT User	If using Windows Authentication, this is the name of the Windows NT user for this process. Also appears as the Windows NT user name if using a trusted connection.
Host	The machine name that originated the process.
Program	The name of the application program that initiated the process.
Memory	Number of pages in the procedure cache (SQL Cache) that are currently allocated to this process. A negative number indicates that pages are being released (freed) from this process.
CPU	Cumulative CPU time for the process. The CPU usage is updated regardless of the value of the SET STATISTICS TIME ON option.
Physical I/O	The current cumulative number of reads and writes issued by the process.
Blocked	If the process is being blocked, this value contains the SPID of the blocking process.
Database	The database where the process is running.
Command	The command the process is currently issuing.
Last Batch	For a client process, this value represents the last time a remote stored procedure call or an EXECUTE statement was executed. For system processes, it represents the time at which SQL Server was started.
Login Time	For client process, this value represents the last time a client logged into the SQL Server instance. For system processes, it represents the time at which SQL Server was started.
Wait Time	Represents the number of milliseconds that the process has been waiting to be serviced. If the process is not waiting, a zero (0) is displayed.
Wait Type	The last or current SQL Server wait type.
Open Xacts	The current number of open transactions owned by the process.
NT Domain	If using Windows Authentication or a trusted connection, this value contains the name of the Windows Domain of the user who owns the process.
Net Address	The unique identifier of the network card in the client machine that owns the process.

Metrics

The key to spotting problem processes is to view their information in light of the total activity on a server. For example, if your database server does not have an overabundance of memory, you should periodically check to see who your heavy memory users are. You should also check the total percentage of memory each takes up. If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing and take action if necessary.

All Locks Tab

- [Metrics](#)

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, SQL Server itself issues lock requests to carry out its internal duties. The All Locks tab gives information the locks currently on the system and also indicates if any blocking situations are occurring. The table below describes the information available on the All Locks tab of the Users Detail view:

Information	Description
SPID	The process id of the process holding the lock.
Login	The login name of the process.
NT User	The operating system name of the process.
Database	The database in which the process is running.
Table Name	The name of the table involved in a lock. This will be NULL for non-table locks or table locks that take place in the tempdb database.
Ndx ID	The index ID involved in the lock.
Lock Type	The type of the lock (database, table, row id, etc.)
Lock Mode	The lock's mode (shared, exclusive, etc.)
Lock Status	The lock's status (waiting or granted).
Lock Owner Type	Whether the lock came from a regular session or a transaction.
User Program	The executable the process is using against the server.
BLK SPID	If zero, the process is not being blocked. If non-zero, this column represents the process ID of the process blocking the requested lock.
Wait Time	The current amount of wait time for the process, in milliseconds.
SPID Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
SPID Command	The command the process is currently issuing.
NT Domain	The name of Windows 2000/NT domain.

Metrics

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the stranglehold on the objects the user was accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

Blocking Locks Tab

- [Metrics](#)
- [Troubleshooting](#)

Metrics

Without a doubt, blocking lock situations can give the appearance of a frozen database almost more than anything else can. A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although SQL Server supports row-level locking, blocking lock situations do crop up - sometimes frequently.

The Blocking Locks tab contains information relating to user accounts that are currently blocked and the sessions that are blocking them. The table below describes the information available on the Blocking Locks tab of the Users Detail view:

Column	Description
SPID	The process id of the process holding the lock.
Login	The login name of the process.
NT User	The operating system name of the process.
BLK SPID	If zero, the process is not being blocked. If nonzero, this column represents the process ID of the process blocking the requested lock.
Database	The database in which the process is running.
Table Name	The name of the table involved in a lock. This will be NULL for non-table locks or table locks that take place in the tempdb database.
Ndx ID	The index ID involved in the lock.
Lock Type	The type of the lock (database, table, row id, etc.)
Lock Mode	The lock's mode (shared, exclusive, etc.)
Lock Status	The lock's status (waiting or granted).
Owner Type	Whether the lock came from a regular session or a transaction.
Program	The executable the process is using against the server.
Wait Time	The current amount of wait time for the process, in milliseconds.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Command	The command the process is currently issuing.
NT Domain	The name of Windows 2000/NT domain.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

Contention Page Statistics

The Contention performance category view displays the following vital SQL Server contention statistics:

- [Blocked Users](#)
- [CPU Busy](#)
- [I/O Busy](#)
- [Latch Waits](#)
- [Lock Overview](#)
- [Log Flush Waits](#)
- [Page Deadlocks](#)
- [Server Idle](#)
- [Table Lock Escalations](#)

Blocked Users

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although SQL Server supports flexible locking mechanisms, blocking lock situations do crop up. User processes holding exclusive locks and not releasing them via a proper COMMIT frequency most often cause blocks.

The Blocked Users statistic displays the number of current process blocked by other processes. Unless a process times out via an application timeout mechanism, or the process has specified a timeout period via the SET LOCK_TIMEOUT command, a process waiting for a lock waits indefinitely.

TIP: Click this statistic to drill down to the [Blocking Lock tab](#) of the Locks view.

Metrics

Consistently seeing positive numbers for the blocked statistic should also clue you into the fact that a bottleneck exists for some processes. You can easily drill down and discover the exact process(es) holding locks that are blocking out other user activity.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects he or she was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

Page Deadlocks

- [Metrics](#)

- [Troubleshooting](#)

The Page Deadlocks statistic represents the number of deadlocks per second detected by SQL Server. Page Deadlocks occur when processes cannot proceed because they are waiting on a set of resources held by each other or held by other processes.

Metrics

Consistently seeing page deadlock counts greater than zero indicates that some user processes are experiencing delays completing their work. When SQL Server identifies a page deadlock, it resolves the situation by choosing the process that can break the deadlock. This process is termed the deadlock victim. SQL Server rolls back the deadlock victim's transaction, and then notifies the process' application by returning an error message. It also cancels the process' request and allows the transactions of the remaining processes to continue.

SQL Server always attempts to choose the least expensive thread running the transaction as the deadlock victim.

Troubleshooting

Because SQL Server automatically resolves deadlock situations, you should work proactively to prevent them in the first place.

The culprit of most blocking lock and deadlock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

You can change default deadlock behavior by using the SET DEADLOCK_PRIORITY command, which reprioritizes a process' position in a deadlock situation.

Number of Deadlocks

The number of deadlocks statistic is the measure of the lock requests per second that resulted in a deadlock.

The culprit of most blocking lock and deadlock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

You can change default deadlock behavior by using the SET DEADLOCK_PRIORITY command, which reprioritizes a process' position in a deadlock situation.

Latch Waits

- [Metrics](#)

The Latch Waits statistic represents the number of latches per second that could not be satisfied immediately by SQL Server. Latches can be thought of as lightweight, mini-locks that are used to protect actions and resources used inside transactions.

TIP: Click this statistic to drill down to the [Waits tab](#) of the Contention Detail view.

Metrics

Unless accompanied by long wait times, latch waits should not pose too much of a performance problem in normal SQL Server installations.

Log Flush Waits

- [Metrics](#)
- [Troubleshooting](#)

The Log Flush Waits statistic represents the number of commits per second waiting for a log flush. SQL Server records every modification to data in a database's log cache. After a transaction is committed, SQL Server writes everything from the database's log cache to the physical log files. Before a process invoking the COMMIT can continue work, it must wait until a log flush finishes. Reduced performance results when a process must wait for a log flush to complete.

TIP: Click this statistic to drill down to the [Waits tab](#) of the Contention Detail view.

Metrics

Log flush waits with no measurable wait time are not considered worrisome. Waits accompanied by considerable wait times are an indicator of contention that degrades overall performance.

Troubleshooting

Drilling down into the log flush detail quickly identifies databases that are experiencing log flush waits. Placing these logs on faster physical devices (and separating them from their parent database) should reduce wait activity.

Table Lock Escalations

- [Metrics](#)

The Table Lock Escalations statistic represents the number of times locks on a table were escalated.

TIP: Click this statistic to drill down to the [All Lock tab](#) of the Locks view.

Metrics

Many table lock escalations could indicate contention problems. If increasing numbers of table lock escalations are viewed at the same time as blocking or deadlock problems, the application design could be at fault.

Lock Overview

- [Metrics](#)
- [Troubleshooting](#)

The Lock Overview tab displays the total number of acquired locks and blocking locks being experienced per database.

Metrics

Consistently seeing positive numbers for the blocked statistic should also clue you into the fact that a bottleneck exists for some processes. You can easily drill down and discover the exact process(es) holding locks that are blocking out other user activity.

Another situation to look for with respect to locking is when the total number of acquired locks reaches the maximum lock limit currently set on SQL Server.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

If the total amount of acquired/requested locks has exceeded the maximum allowed on SQL Server, you should increase the configuration parameter locks.

CPU Busy

- [Metrics](#)

The CPU Busy statistic represents the time in milliseconds that the CPU has spent working since the last refresh in Performance Center.

Metrics

None.

I/O Busy

- [Metrics](#)

The Server I/O Busy statistic represents the time in milliseconds that SQL Server has spent performing input and output operations since the last refresh in Performance Center.

Metrics

None.

Server Idle

- [Metrics](#)

The Server Idle statistic represents the time in milliseconds that SQL Server has spent idle since the last refresh in Performance Center.

Metrics

None.

Contention Detail View

The following tabbed pages are available on the Contention Detail view:

- [Blocking Locks](#)
- [Locks](#)
- [Waits](#)

Locks Tab

- [Metrics](#)

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, SQL Server itself issues lock requests to carry out its internal duties. The Locks tab of the Contention Detail view displays two grids that contain information about all locks currently on the system. The [first grid](#) displays an overview of lock activity by database. The [second grid](#) displays a detailed view of lock activity.

The table below describes the information available in the Lock Summary grid on the Locks tab of the Contention Detail view:

Column	Description
Database	The database containing the locks.
Lock Type	The type of lock acquired.
Lock Count	The number of locks for the database/lock type combination.

The table below describes the information available in the Lock Detail grid on the Locks tab of the Contention Detail view:

Column	Description
SPID	The process id of the process holding the lock.
Login	The login name of the process.
NT User	The operating system name of the process.
Database	The database where the locks are occurring.
Table Name	The name of the table involved in a lock. This will be NULL for non-table locks or table locks that take place in the tempdb database.
Ndx ID	The index ID involved in the lock.
Lock Type	The type of lock (database, table, row id, etc.)
Lock Mode	The mode of the lock (shared, exclusive, etc.)
Lock Status	The status of the lock (waiting or granted).
Owner Type	Whether the lock came from a regular session or a transaction.
Program	The executable the process is using against the server.
BLK SPID	If nonzero, the process ID of the process blocking the requested lock. If zero, the process is not being blocked.
Wait Time	The amount of time the process has waited for the lock, in milliseconds.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Command	The command the process is currently issuing.
NT Domain	The name of the Windows 2000/NT domain.

NOTE: The information in the Lock Detail grid is available in the Lock Detail grid on the Locks tab and the [All Locks tab](#) of the Locks view.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

Locks that are held for unusually long periods could be candidates for further investigation. The application logic could be inefficient or perhaps the program is not issuing frequent enough COMMITs.

Blocking Locks Tab

- [Metrics](#)
- [Troubleshooting](#)

Without a doubt, blocking lock situations can give the appearance of a frozen database almost more than anything else. A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although SQL Server supports row-level locking, blocking lock situations do crop up - sometimes frequently.

Metrics

The Blocking Locks tab of the Contention Detail view displays two grids that contain information about all blocking lock situations on the system. The [first grid](#) displays the database name and count of all blocking locks inside the database. The [second grid](#) contains information relating to user accounts that are currently blocked and the sessions that are blocking them.

The table below describes the information available in the Blocking Lock Summary grid on the Blocking Locks tab of the Contention Detail view:

Column	Description
Database	The database containing the locks.
Locks Blocked	The number of blocked locks.

The table below describes the information available in the Blocking Lock Detail grid on the Blocking Locks tab of the Contention Detail view:

Column	Description
SPID	The process ID of the process holding the lock.
Login	The login name of the process.
NT User	The operating system name of the process.
BLK SPID	If nonzero, the process ID of the process blocking the requested lock. If zero, the process is not being blocked.
Database	The database where the locks are occurring.
Table Name	The name of the table involved in a lock. This will be NULL for non-table locks or table locks that take place in the tempdb database.
Ndx ID	The index ID involved in the lock.
Lock Type	The type of lock (database, table, row id, etc.)
Lock Mode	The mode of the lock (shared, exclusive, etc.)
Lock Status	The status of the lock (waiting or granted).
Owner Type	Whether the lock came from a regular session or a transaction.
Program	The executable the process is using against the server.
Wait Time	The amount of time the process has waited for the lock, in milliseconds.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Command	The command the process is currently issuing.
NT Domain	The name of the Windows 2000/NT domain.

NOTE: The information in the Blocking Lock Detail grid is available in the Blocking Lock Detail grid on the Locks tab and the [Blocking Locks tab](#) of the Locks view.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

Waits Tab

- [Metrics](#)

Understanding where SQL Server is delayed in its processing can yield big benefits in increasing overall response time. The Waits tab of the Contention Detail view contains two grids that display wait information. The first grid is a summary of various wait type categories, along with their overall wait counts and wait times. The second grid displays individual wait statistics, along with their individual wait counts and wait times.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

Waits could not be causing actual delays in SQL Server processing unless they are accompanied by substantial wait times. The ones displaying high wait times are the ones likely to be causing transactional and system slowdowns.

Database Page Statistics

The Database view includes the following statistics:

- [Database Summary](#)
- [Databases](#)
- [Databases Needing Backup](#)
- [Databases Without Auto-Create Statistics](#)
- [Databases Without Auto-Growth](#)
- [Databases Without Auto-Update Statistics](#)
- [Errors in Current Error Log](#)
- [Files](#)
- [File Groups](#)
- [Logs Without Auto-Growth](#)
- [Offline Databases](#)
- [Suspect Databases](#)

Related Topics

[Database Detail View](#)

[Home View Statistics](#)

[Contention Statistics](#)

[I/O Page Statistics](#)

[Memory Page Statistics](#)

[Network Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

[Users Page Statistics](#)

Errors in Current Error Log

- [Metrics](#)
- [Troubleshooting](#)

SQL Server records various system events in its system or error log. The majority of messages placed into the log are informational in nature, however since some contain critical messages, you should immediately review them. This section indicates the number of actual error messages in the most recent error log so you know if there are potential events that require your attention.

TIP: Click this statistic to open the [I/O performance category view](#).

Metrics

You should investigate any positive values for errors.

Troubleshooting

If non-zero values are found for this statistic, you should drill down into the current error log and examine the detail found that accompanies each error issued by SQL Server.

Databases

- [Metrics](#)

The Database statistic represents total number of databases defined on a SQL Server.

TIP: Click this statistic to drill down to the [SQL Cache tab](#) of the Memory Detail view.

Metrics

None.

Files

- [Metrics](#)

The Files statistic represents total number of files (both database and log) defined on a SQL Server.

TIP: Click this statistic to open the [Memory performance category view](#).

Metrics

None.

File Groups

- [Metrics](#)

The File Groups statistic represents total number of file groups (that contain both database and log files) defined on a SQL Server.

TIP: Click this statistic to open the [Memory performance category view](#).

Metrics

None.

Suspect Databases

- [Metrics](#)
- [Troubleshooting](#)

The Suspect Databases statistic represents the number of databases SQL Server has marked as suspect. Databases are marked suspect by SQL Server if they fail during automatic recovery, which is performed at server startup. If serious damage is experienced by a database during regular uptime, SQL server will also mark a database as suspect.

Metrics

You should not find any suspect databases on any production server. You should immediately investigate any non-zero numbers, for this statistic.

Troubleshooting

The steps to handling a suspect database will vary from one installation to another. However, here are some general guidelines you can use to troubleshoot a suspect database:

- Begin by examining the SQL Server error log for clues as to what caused the database to be marked as suspect.
- It is not unusual for a server to run out of physical disk space on drives used by SQL Server. When this happens, recovery for databases can sometimes fail with the end result being SQL Server marking a database as suspect. To remedy this situation, you should free up space on the identified drives or add files to the newly marked suspect database. For SQL Server 2000, this can be accomplished by utilizing the following stored procedures: `sp_add_data_file_recover_suspect_db` and `sp_add_log_file_recover_suspect_db`. For version 7.0 of SQL Server, you will need to use the `sp_resetstatus` stored procedure to reset the suspect status flag for the database in question, use the `alter database` command to add new datafiles to the database, and then stop/start the SQL Server.
- Many times, suspect databases are caused by SQL Server not being able to access a database or log file. This happens if a particular physical hard drive has become unavailable, but also can occur if another operating system process has obtained exclusive access to a file. If this scenario proves to be true, once you have ensured that the file(s) are available once again to the operating system, you can use the `sp_resetstatus` stored procedure to reset the suspect status flag for the database and then stop/start the SQL Server.

If none of these solutions are possible, you will likely have to restore your database using the last full and transaction log backups.

Offline Databases

- [Metrics](#)
- [Troubleshooting](#)

The Offline Databases statistic represents the number of databases SQL Server has offline, meaning that no database modifications can occur.

TIP: Click this statistic to open the [I/O performance category view](#).

Metrics

You should not find any offline databases on any production server. You should immediately investigate any non-zero numbers, for this statistic.

Troubleshooting

Should an offline database be found by Performance Center, you can easily place it back online by utilizing either the `sp_dboption` stored procedure or the `alter database` command.

Databases Needing Backup

- [Metrics](#)
- [Troubleshooting](#)

The Databases Needing Backup statistic represents the number of databases in SQL Server that have not been backed up for more than seven days. This statistic excludes the pubs, tempdb, Northwind, msdb, and model databases.

TIP: Click this statistic to open the [I/O performance category view](#).

Metrics

To ensure proper protection for most databases, it is recommended that even the most static databases be backed up at least once a week. You should frequently backup critical, dynamic databases and include transaction log backups to enable point-in-time recovery ability.

Troubleshooting

Any critical databases found with obsolete or no backups should immediately be backed up. Moreover, to ensure proper data protection for each database, you should institute a planned backup schedule. The timing and repetition of the backups depend on the critical recovery needs of each database.

Databases Without Auto-Growth

- [Metrics](#)
- [Troubleshooting](#)

Beginning in SQL Server 7.0, a DBA was given the ability to tell SQL Server to automatically grow a database in size when more space was required. This feature can save a critical transaction or other database request from failing due to a lack of free space in the database. It is recommended that critical databases have this feature enabled.

The Databases Without Auto-Growth statistic provides a count of databases that do not have their automatic growth property enabled.

TIP: Click this statistic to drill down to the [Blocking Lock tab](#) of the Locks view.

Metrics

Static databases (those not expected to grow in size) will likely not need their auto-growth property enabled. Growing, dynamic databases should almost always be allowed to automatically grow when needed.

Troubleshooting

If any critical, dynamic database is found to not have its auto-growth feature enabled, then the DBA can modify the file(s) used by the database to automatically grow by using the ALTER DATABASE ... MODIFY FILE ... FILEGROWTH command. You should also ensure that the MAXSIZE setting of each file is set appropriately.

Logs Without Auto-Growth

- [Metrics](#)
- [Troubleshooting](#)

Beginning in SQL Server 7.0, a DBA was given the ability to tell SQL Server to automatically grow a database or transaction log in size when more space was required. This feature can save a critical transaction or other database request from failing due to a lack of free space in the database or transaction log. It is recommended that critical databases and their transaction logs have this feature enabled.

The Logs Without Auto-Growth statistic provides a count of transaction logs that do not have their automatic growth property enabled.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

Static databases (those not expected to grow in size) will likely not need their transaction log's auto-growth property enabled. Growing, dynamic databases should almost always have their transaction log be set to automatically grow when needed.

Troubleshooting

If any critical, dynamic database is found to not have their transaction log auto-growth feature enabled, then the DBA can modify the file(s) used by the database's transaction log to automatically grow by using the ALTER DATABASE ... MODIFY FILE ... FILEGROWTH command. You should also ensure that the MAXSIZE setting for each file is set appropriately.

Databases Without Auto-Update Statistics

- [Metrics](#)
- [Troubleshooting](#)

The Databases Without Auto-Update Statistics statistic represents the total number of databases defined on SQL Server that do not have the AUTO_UPDATE_STATISTICS option enabled.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

When the AUTO_UPDATE_STATISTICS option is enabled, SQL Server automatically updates existing statistics when the statistics become out-of-date because data in the tables has changed enough to affect the optimizer's decision-making process.

Troubleshooting

If possible, a DBA should keep databases in AUTO_UPDATE_STATISTICS mode. If a database is found without this option set, you can easily change its auto-update statistics to true by using the command:

```
EXEC sp_dboption '(database name)',
```

Databases Without Auto-Create Stats

- [Metrics](#)
- [Troubleshooting](#)

The Databases Without Auto-Create Stats statistic represents the total number of databases defined on SQL Server that do not have the AUTO_CREATE_STATISTICS option enabled.

TIP: Click this statistic to open the [Contention performance category view](#).

Metrics

When the AUTO_CREATE_STATISTICS option is enabled, statistics are automatically created on columns used in a SQL query predicate. Keeping object statistics fresh in the SQL Server data dictionary improves query performance because the optimizer can better determine how to evaluate a query and return the requested data. If the statistics are not used, SQL Server should automatically delete them.

Troubleshooting

If possible, a DBA should keep their databases in AUTO_CREATE_STATISTICS mode. If a database is found without this option set, you can easily change it by using the command:

```
EXEC sp_dboption '<database name>', 'auto create statistics', true
```

Database Summary

- [Metrics](#)
- [Troubleshooting](#)

The Database Summary gives a bird's eye view of situations that could require a DBA's attention.

TIP: Click the table heading to drill down to the [Overview tab](#) of the Database Detail view.

Metrics

You should investigate any blocking lock situations and databases that are using an inappropriate amount of memory, CPU, or I/O. In addition, you should reserve off-hours submissions for various utilities such as DBCCs, BCPs, or backup and restore operations.

Troubleshooting

If a blocking lock situation is observed, you should drill down and find the blocking locks that exist using Performance Center's Lock/Blocking Locks view. Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds a process waits for a lock before timing out.

Database Detail View

The Database view Detail includes the following tabbed pages:

- [Backup History](#)
- [Index Details](#)
- [Options](#)
- [Overview](#)
- [Replication](#)
- [SQL Agent](#)
- [Table Details](#)

Overview Tab

- [Metrics](#)
- [Troubleshooting](#)

The Overview tab of the Database Detail view displays summarized metadata for all defined databases on SQL Server. The table below describes the information available on the Overview tab of the Database Detail view:

Information	Description
Database	The name of the database.
Created	The date/time when the database was created.
Status	The status of the database (online, offline, suspect, etc.)
DB Size (MB)	The total size of the database in MB.
Log Size (MB)	The total size of the log in MB.
Tables	The number of all tables in the database.
Indexes	The number of all indexes in the database.
Users	The number of all defined user accounts in the database.
Last Full Backup	The date/time of the last full backup for the database.

Metrics

The following are things to take note of:

- Any critical database that shows a last full backup date that is older than the database's required backup needs.
- Any database that shows a status of offline or suspect.
- Any growing database that does not have its database or log files set to automatically grow in size.
- Any dynamic database that does not have its object statistics set to automatically update.

Troubleshooting

Depending on the situation, you should take the following actions:

For databases that require a full backup, perform a full backup of the database when appropriate.

For suspect databases, the steps to handling will vary from one installation to another. However, here are some general guidelines you can use to troubleshoot a suspect database:

- Begin by examining the SQL Server error log for clues as to what caused the database to be marked as suspect.
- It is not unusual for a server to run out of physical disk space on drives used by SQL Server. When this happens, recovery for databases can sometimes fail with the end result being SQL Server marking a database as suspect. To remedy this situation, you should free up space on the identified drives or add files to the newly marked suspect database. For SQL Server 2000, this can be accomplished by utilizing the following stored procedures: `sp_add_data_file_recover_suspect_db` and `sp_add_log_file_recover_suspect_db`. For SQL Server version 7.0, you will need to use the `sp_resetstatus` stored procedure to reset the suspect status flag for the database in question, use the `alter database` command to add new datafiles to the database, and then stop/start SQL Server.
- Many times, suspect databases are caused by SQL Server not being able to access a database or log file. This happens if a particular physical hard drive has become unavailable, but also can occur if another operating system process has obtained exclusive access to a file. If this scenario proves to be true, once you have ensured that the file(s) are available once again to the operating system, you can use the `sp_resetstatus` stored procedure to reset the suspect status flag for the database and then stop/start SQL Server.

Should an offline database be found by Performance Center, you can easily place it back online by utilizing either the `sp_dboption` stored procedure or the `alter database` command.

If any critical, dynamic database or log is found to not have their auto-growth feature enabled, then the DBA can modify the file(s) used by the database to automatically grow by using the `ALTER DATABASE ... MODIFY FILE ... FILEGROWTH` command. You should also ensure that the `MAXSIZE` setting for each file is set appropriately.

If possible, a DBA should keep their databases in `AUTO_CREATE_STATISTICS` and `AUTO_UPDATE_STATISTICS` mode. If a database is found without this option set, you can easily change it by using the following commands:

```
EXEC sp_dboption '<database name>', 'auto create statistics', true
EXEC sp_dboption '<database name>', 'auto update statistics', true
```

SQL Agent Tab

SQL Server provides the ability to submit and run jobs as well as be notified about certain SQL Server-related events. The SQL Agent tab of the Database Detail view displays information regarding the status of the server's SQL Agent as well as details for all jobs and alerts that are defined to the system.

SQL Server Job Summary

- [Metrics](#)

The SQL Server Job Summary section graphically displays the outcome of all jobs for the last 25 job outcomes.

Metrics

You should investigate any counts noted for failed jobs.

SQL Server Alert Summary

The SQL Server alert summary section graphically displays the number of times that performance and event alerts have fired.

Backups History Tab

The Backups tab of the Database Detail includes the following:

- [Backup Detail](#)

Backup Detail

- [Metrics](#)

The Backup History tab of the Database Detail view displays the most recent 25 backups for a selected database. The table below describes the information available on the Backup History tab of the Database Detail view:

Information	Description
Database	The name of the database.
Backup Start	The time stamp when the backup began.
Backup Finish	The time stamp when the backup finished.
Backup Type	The type of backup (FULL, INCREMENTAL, etc.)
Backup Size (KB)	The size of the backup, in kilobytes.
Expiration Date	The expiration date for the backup, if any.

Metrics

Although the needs of an application determine the frequency and type of backup, it is generally recommended that most dynamic databases have a solid plan in place for full and differential backups. For databases requiring point-in-time recovery, a backup plan should also include log backups.

Replication Tab

The Replication tab of the Database Detail view provides basic information regarding replication activities that are occurring on the monitored SQL Server. The replication tab includes the following sections:

- [Replication Agent Details](#)
- [Replication Throughput Details](#)

The [first grid](#) displays information about replication agent activities. The [second grid](#) delivers information regarding replication throughput for the server.

The table below describes the information available in the Agent Information grid on the Replication tab of the Database Detail view

Information	Description
Agent	The agent performing the replication work.
Publication	The object being published to a subscribing database.
Publisher	The SQL Server providing the source material.
Publisher DB	The database of the SQL Server providing the source material.
Status	The current status of the replication agent's task.
Subscriber	The SQL Server requesting the source material.
Subscriber DB	The database of the SQL Server requesting the source material.
Type	The type of replication activity (push, pull, etc.)
Start Time	The start time for the replication task.
Duration	The duration of the replication task.
Last Action	The output message for the replication task.

The table below describes the information available in the SQL Cache Details grid on the Replication tab of the Database Detail view:

Information	Description
Owner	The owner of the table.
Table Name	The name of the table.
Pinned	This indicates if the table is pinned in memory.
File Group	The filegroup name that contains the table.
Rows	The number of rows in the table.
Clustered	This indicates if the table has a clustered index.
Indexes	This provides a count of the number of indexes for the table.
FK Constraints	This indicates if the table has any defined foreign keys.
Chk Constraints	This indicates if the table has any defined check constraints.
Has Identity	This indicates if the table has an identity column.

NOTE: For space-related table information, use the [Tables tab](#) of the Space Detail view.

Index Details Tab

- [Metrics](#)
- [Troubleshooting](#)

The Index Details tab of the Database Detail view provides an overview of the attributes of every user index defined in the database. The table below describes the information available on the Index Details tab of the Database Detail view:

Information	Description
Table Name	The name of the table.
Index Name	The name of the index.
File Group	The filegroup name that contains the index.
Clustered	This indicates if the index is clustered.
Unique	This indicates if the index is unique.
Depth	This indicates the depth/level of the index.
Auto-Stats	This indicates if statistics are automatically recomputed for the index.
Max Nonleaf Index Row Sizes	This indicates the longest size of a non-leaf index row.
Max Length	This indicates the maximum size of a row.

NOTE: For space-related index information, use the [Indexes tab](#) of the Space Detail view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Seeing an index with a depth of more than three could indicate the index has become fragmented and is need of a rebuild. Indexes built on dynamic objects should have their statistics automatically collected by SQL Server to ensure the optimizer has the most up-to-date information to work with when making access path decisions.

Troubleshooting

Performance Center offers several options to rebuild an index:

- 1 You can use DBCC DBREINDEX to rebuild all the indexes for a selected table or selectively rebuild just the indexes you want.
- 2 For SQL Server 2000 and later, you can use DBCC INDEXDEFRAG to rebuild the indexes for a table online. Users can perform DML operations on the objects involved in the DBCC operation.
- 3 You can drop and re-create the indexes that appear fragmented.

For indexes that are not having statistics automatically gathered by SQL Server, you can drop and recreate the index with the option of having statistics recomputed by the server.

Table Details Tab

- [Metrics](#)

The Table Details tab of the Database Detail view provides an overview of the attributes of every user table defined in the database. The table below describes the information available on the Table Details tab of the Database Detail view:

Column	Description
Owner	The owner of the table.
Table Name	The name of the table.
Pinned	Indicates if the table has been pinned in the buffer cache.
File Group	The file group where the table resides.
Rows	The number of rows in the table (could be inaccurate if statistics are not kept up to date).
Clustered	Whether the table has a clustered index defined or not.
Indexes	The number of indexes defined to the table.
FK Constraints	Whether the table has foreign key constraints defined to it.
Chk Constraints	Whether the table has check constraints defined to it.
Identity	Whether the table has an IDENTITY column defined to it

NOTE: For space-related table information, use the [Tables tab](#) of the Space Detail view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

None.

Network Statistics

The Network view displays the following vital network statistics:

- [Network Bytes Read](#)
- [Network Bytes Written](#)
- [Network I/O Wait Requests](#)
- [Network I/O Wait Time](#)
- [Network Reads](#)
- [Network Writes](#)
- [Packet Errors](#)
- [Packets Received](#)
- [Packets Sent](#)
- [Total Traffic](#)

Related Topics

[Home View Statistics](#)

[Database Statistics](#)

[Contention Statistics](#)

[I/O Statistics](#)

[Memory Statistics](#)

[Space Statistics](#)

[Users Statistics](#)

Packets Received

- [Metrics](#)

SQL Server counts incoming packets from the time the server is initiated. This statistic displays the delta count of all incoming packets received over the network since the last Performance Center sampling.

Metrics

None.

Packets Sent

- [Metrics](#)

SQL Server counts outgoing packets from the time the server is initiated. This statistic displays the delta count of all outbound packets written to the network since the last Performance Center sampling.

Metrics

None.

Network Reads

- [Metrics](#)

The number of network reads performed by SQL Server since the last Performance Center sampling.

Metrics

None.

Network Writes

- [Metrics](#)

The number of network writes performed by SQL Server since the last Performance Center sampling.

Metrics

None.

Network Bytes Read

- [Metrics](#)

The number of bytes that SQL Server read from the network since the last Performance Center sampling.

Metrics

None.

Network Bytes Written

- [Metrics](#)

The number of bytes that SQL Server wrote to the network since the last Performance Center sampling.

Metrics

None.

Packet Errors

- [Metrics](#)
- [Troubleshooting](#)

SQL Server counts packet errors from the time the server is first initiated. This statistic displays the delta count of all packet errors that occurred since the last Performance Center sampling.

Metrics

This value should remain low relative to your overall network stability and efficiency. Networks with great amounts of congestion and other traffic problems causes these errors to occur.

Troubleshooting

In the event that packet errors continues to remain elevated beyond what you would expect in your particular environment, contact a Network Administrator for assistance in tracking the problem.

Total Traffic

- [Metrics](#)

The total number of packets that SQL Server received and sent over the network since the last Performance Center sampling.

Metrics

None.

Network I/O Wait Requests

- [Metrics](#)

The total number of times that SQL Server generated an I/O wait request since the last Performance Center sampling.

Metrics

None.

Network I/O Wait Time

- [Metrics](#)
- [Troubleshooting](#)

The number of milliseconds that SQL Server remained in a wait state due to Network I/O Wait Requests since the last Performance Center sampling.

Metrics

None.

OS Page Statistics

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Center lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX

NOTE: To view processor info and swap disk info on an HP-UX box, you need to login as ROOT in the OS login.

- Linux
- Solaris
- Unix
- Windows XP, 2000, and NT

Summary Tab

The OS Summary tab displays the following statistics to communicate the general overall performance levels of the operating system:

- [Disk Time](#)
- [Load Average](#)
- [Processor Time](#)
- [Paged Memory Used \(Windows\)](#)
- [Swap Memory Used \(AIX, HP-UX, Linux, Solaris, Unix\)](#)

- [Average Disk Queue](#)
- [Network Output Queue \(Windows\)](#)
- [Network Queue \(Solaris\)](#)
- [Page Faults/Sec](#)
- [Processor Queue](#)
- [Processor Speed](#)
- [Processor](#)
- [Available Paged Memory \(Windows\)](#)
- [Available Physical Memory](#)
- [Available Swap Memory \(AIX, HP-UX, Linux, Solaris, Unix\)](#)
- [Total Paged Memory \(Windows\)](#)
- [Total Physical Memory](#)
- [Total Swap Memory \(AIX, HP-UX, Linux, Solaris, Unix\)](#)
- [Free Disk Space](#)
- [Total Disk Space](#)
- [Used Disk Space](#)
- [Number of Logins](#)
- [Number of Processes](#)
- [Number of Processors](#)
- [Top CPU Process](#)
- [Top I/O Process](#)
- [Top Memory Process](#)

Processor Time

The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

Metrics

If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.

Troubleshooting

Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

Processor Speed

The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Processor

The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Disk Time

The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

Metrics

You should avoid consistently seeing values for this statistic greater than 90%.

Troubleshooting

Add more disk drives and partition the files among all of the drives.

Load Average

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Metrics

High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Paged Memory Used

The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

NOTE: This statistic is available for the Windows platform.

Metrics

This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.

Troubleshooting

Increase the size of page file.

Number of Processors

This statistic displays the number of processors currently in use.

Swap Memory Used

The Swap Memory Used statistic is the percentage of swap space currently in use.

Metrics

If the percentage of swap memory used is above 90%, you may be running out of memory.

Troubleshooting

Increase the size of your swap files.

Average Disk Queue

The Average Disk Queue statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Metrics

This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.

Troubleshooting

Some things you can do if you have problems with this statistic include:

- Redistribute the data on the disk with the large average disk queue to other disks.
- Upgrade to faster disk(s).

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics

This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue

The Processor Queue Length statistic is the number of threads in the processor queue.

Metrics

Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting

A sustained high value in the Processor Queue could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Network Output Queue/Network Queue

The Network Output Queue Length statistic is the number of threads in the processor queue.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics

Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting

A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Available Physical Memory

The Available Physical Memory statistic represents the amount of RAM available to all processes.

Metrics

This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Troubleshooting

Some things you can do if you have problems with this statistic include:

- Check the running processes to see if there are any memory leaks.
- Stop any services that are not required.
- Install additional RAM.

Available Paged Memory

The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

NOTE: This statistic is available for the Windows platform.

Metrics

If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.

Troubleshooting

Increase the size of page file.

Available Swap Memory

The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Metrics

If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.

Troubleshooting

Increase the size of swap files.

Total Physical Memory

The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Metrics

This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Total Paged Memory/Total Swap Memory

The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics

It is recommended that this value is between 1.5 to 3 times the amount of RAM on the system.

Used Disk Space

The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Total Disk Space

Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two:

- 1 If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- 2 You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Free Disk Space

The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Top Memory Process

Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics

If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Processes Overview

The Processes Overview of the OS Summary includes the following sections:

- [Top CPU Process](#)
- [Top I/O Process](#)
- [Top Memory Process](#)

Top CPU Process

Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics

If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.

Troubleshooting

Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process

The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Number of Logins

This statistic displays the total number of logins on the server.

Number of Processes

This statistic displays the total number of processes on the server.

CPU Tab

The CPU tab of the OS Detail includes the following sections:

- [Context Switches/Sec](#)
- [CPU Utilization](#)
- [Interrupts/Sec](#)
- [Processor Queue Length](#)

CPU Utilization

The CPU Utilization section includes the following information:

- [% Privileged Time](#)
- [% User Time](#)

% Privileged Time

The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

NOTE: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

Metrics

The ideal range should be 0-40% (less than 40% indicates excessive system activity).

Troubleshooting

If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

% User Time

The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Metrics

If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

Interrupts/Sec

The Interrupts/Sec statistic is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. It does not include deferred procedure calls (DPCs), which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every ten milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Metrics

The ideal range should be 0-5000. A number greater than 5000 indicates possible excessive hardware interrupts; justification is dependent on device activity.

Context Switches/Sec

The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Metrics

The ideal range should be between 0-10,000. GA number greater than 10,000 may indicate too many threads contending for resources.

Processor Queue Length

The Processor Queue Length statistic is the number of threads in the processor queue. There is a single queue for processor time even on computers with multiple processors.

NOTE: For Windows systems, unlike the disk counters, this counter shows ready threads only, not threads that are running.

Metrics

A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Processes Tab

The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

Column	Description
Process	The name of the process.
User	The user of the process.
ID	The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
CPU	The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.
User Mode	The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.
Memory WINDOWS ONLY	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory (MB)	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory	Memory is the percentage of the memory used of the total memory.
Active Memory	Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.
I/O Data	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
Elapsed Time	The total elapsed time, in seconds, that this process has been running.
Thread Count	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.

Column	Description
Handle Count	The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.
Priority	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Creating Proc ID	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Page Faults/Sec	Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.
Page File	Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private	Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.

I/O Tab

The table below describes the information available in this section:

Column	Description
Disk	The disk number assignment.
Reading (KB/s)	The amount of bytes read from the device.
Writing (KB/s)	The amount of bytes written to the device.
Disk Read Time	Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Disk Write Time	Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Disk Time	Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.
Avg. Read Queue	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
Avg. Write Queue	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Disk Reads/Sec	Disk Reads/Sec is the rate of read operations on the disk.
Disk Writes/Sec	Disk Writes/Sec is the rate of write operations on the disk.

Memory Tab

The Memory tab of the OS Detail page includes the following sections:

- [Cache Efficiency](#)
- [Free Physical](#)

- [Free Paged](#)
- [Paging Activity](#)
- [Page Faults](#)
- [Total Physical](#)
- [Total Paged](#)

Paging Activity

The Paging Activity section includes the following statistics:

- [Pages Input/Sec](#)
- [Pages Output/Sec](#)

Pages Input/Sec

The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Metrics

This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Pages Output/Sec

The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Metrics

Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Free Physical

The Free Physical statistic is the amount of physical memory that is uncommitted.

Metrics

None.

Free Paged

The Free Paged statistic is the amount of uncommitted virtual memory.

Metrics

None.

Total Physical

The Total Physical statistic is the total physical memory available.

Metrics

None.

Total Paged

The Total Paged statistic is the amount of total virtual memory, in bytes. Used Memory is the physical memory that has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.

Metrics

None.

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics

This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Cache Efficiency

The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

- [Copy Read Hits%](#)
- [Data Map Hits%](#)
- [MDL Read Hits%](#)
- [Pin Read Hits%](#)

Copy Read Hits %

The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

Metrics

When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Data Map Hits %

The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %

The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %

The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab

The Space tab of the OS Detail page includes the following sections:

- [Disk Space Free](#)
- [Disk Space Detail](#)

Disk Space Free

The Disk Space Free metric displays the amount of free disk space in megabytes.

Metric

None.

Disk Space Detail

The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Network Tab

The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

Column	Description
Network Interface	The name of network interface.
INET Address/Address	The IP address assigned to the network interface.
Pkts Sent/Sec	The number of packets sent per second.
Pkts Received/Sec	The number of packets received per second.
Sent (KB/Sec)	The number of bytes sent per second.
Received (KB/Sec)	The number of bytes received per second.
Out Pkts Discarded	The number of outbound packets discarded.
In Pkts Discarded	The number of inbound packets discarded.
Out Pkt Errors	The number of outbound packet errors.
In Pkt Errors	The number of inbound packet errors.
Queue Length	The queue length.
Collisions	The number of collisions.
Packets Discarded	The number of packets discarded.

Sybase Expert Guide

This section includes expert help for all Sybase ASE categories and statistics in the Performance Center views and pages. For detailed information on using the application, see [Using Performance Center](#). This help is divided into the following sections:

[Home View](#)

[Memory Statistics](#)

[I/O Statistics](#)

[Space Statistics](#)

[Databases Statistics](#)

[Contention Statistics](#)

[Users Statistics](#)

[Network Statistics](#)

[OS Views and Statistics](#)

Home View Statistics - Sybase

The Performance Center Home view lets you review availability and overall performance of all monitored databases from a single window. Statistics on the Home view are organized into the following categories:

- [Contention Vital Signs](#)
- [I/O Vital Signs](#)
- [Memory Vital Signs](#)
- [Network Vital Signs](#)
- [Space Vital Signs](#)
- [Users Vital Signs](#)

Memory Vital Signs

The following memory statistics are on the Sybase Home view:

- [Data Cache Hit Rate](#)
- [Procedure Cache Hit Rate](#)
- [Large I/O Hit Rate](#)
- [Clean Buffer Grab Rate](#)

Data Cache Hit Rate

- [Metrics](#)

- [Troubleshooting](#)

Data read from memory produces user response times many times faster than when that same data is read from disk. The Sybase data cache assists with keep physical I/Os to an absolute minimum.

The data cache hit rate is an excellent indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. The table below describes the three key counters in Sybase used to arrive at this statistic:

Counter	Description
LOGICAL READS	Data read from memory for user requests.
PAGES PER I/O	The number of pages retrieved in a single I/O operation.
PHYSICAL READS	Data read physically from disk.

TIP: Click this statistic to drill down to the [Cache Activity tab](#) of the Memory Detail view.

Metrics

To help ensure excellent performance, keep your cache hit rate in the neighborhood of 90% or greater. Lower amounts can be okay for user ad hoc databases where sporadic, large table scan operations occur. However, anything below this general threshold for normal databases can require tuning attention, and the adjustment of the Sybase memory tuning parameters.

If you are using named caches, you can drill down into the cache hit rates for each named cache. This helps you understand which objects/operations are depressing the overall cache hit rate for the server.

Troubleshooting

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter or reduce the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when reducing the procedure cache, as this could reduce performance in the server as it relates to reading procedures in from disk.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the data cache (and any named caches), but in 12.5, if you want to reduce the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the data cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary and examination of unnecessary large table scan operations. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Procedure Cache Hit Rate

- [Metrics](#)
- [Troubleshooting](#)

The Sybase procedure cache is used to hold the definitions and query plans of stored procedures and triggers. It is used for short-term memory needs like statistics and query plans for parallel queries. When a user executes a stored procedure, Sybase looks in the procedure cache for a query plan to use. If a query plan is available, Sybase places it on the most recently used (MRU) end of the memory chain and the procedure begins to execute. If no execution plan is in memory, or if all copies of the plan are currently being used, the query tree for the procedure is read in again from the data dictionary, optimized, put on the MRU end of the chain, and executed. Note that other operations, like CREATE INDEX, can also use the procedure cache even when no procedure is referenced.

The more often that a procedure's plan and definition can be referenced in memory, the better the procedure execution time.

TIP: Click this statistic to drill down to the [Cache Activity tab](#) of the Memory Detail view.

NOTE:

Metrics

A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for procedure code reference. Note that when a database is first started, the procedure cache hit rate is not at an optimal level because all code being used is relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

You can drill down into the procedure cache to view the procedures currently in memory along with how much memory they are consuming.

If there is not enough memory to load a requested procedure, or the maximum number of compiled objects is already in use, Sybase returns an error (normally a 701).

Troubleshooting

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). You should be careful when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase server, but in 12.5, if you want to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the procedure cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Large I/O Hit Rate

- [Metrics](#)
- [Troubleshooting](#)

Large I/O can be enabled by splitting the default or any named cache into pools. By default, Sybase performs I/O operations based on a 2-KB page size. For queries where pages are stored and accessed in a sequential manner, it is possible to read many more data pages in a single I/O operation. Large I/O can greatly reduce disk access time when the right situations exist. Operations that routinely perform large table scans, access image or text data, do bulk copies, scan the leaf level of nonclustered indexes, or initiate DBCC tasks can benefit from large I/O.

If large I/O has been configured and is being used, you should observe a high percentage of hits (the number of times large I/O could be performed vs. the number of times large I/O requests were denied by the server). If large I/O is not configured, no large I/O activity should be present.

Metrics

As you might expect, if large I/O is in use, a high hit rate is desirable. You should strive for a hit ratio between 90-100%, with 90% being a good performance benchmark.

Troubleshooting

If large I/O is configured, but you see a low hit rate, you should configure more caches for large I/O use.

Clean Buffer Grab Rate

- [Metrics](#)
- [Troubleshooting](#)

As information is requested from users, buffers are moved into and out of the Sybase data cache. Pages are also modified in the cache (termed dirty buffers) and need to be written out to disk. If Sybase has to wait for a dirty buffer to be written out to disk before a requested buffer is placed into the cache, performance can suffer.

The clean buffer grab rate represents the percentage of time clean buffers were found and referenced in the cache as opposed to Sybase finding dirty buffers.

Metrics

Ideally, the clean buffer grab rate should stay at or near 100%.

Troubleshooting

Seeing a poor clean buffer grab rate for either the default or named caches could indicate that the cache size is too small. You can look into adjusting the total memory configuration parameter higher. Keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

- [Suspect Databases](#)

Contention Vital Signs

The following contention statistics are on the Sybase Home view:

- [Blocking Lock Rate](#)
- [Deadlock Rate](#)
- [Device I/O Contention](#)
- [Network Contention Rate](#)

Blocking Lock Rate

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although Sybase supports flexible locking mechanisms, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

The blocking lock rate displays the percentage of times blocks occurred vs. the percentage of locks requested and immediately granted.

TIP: Click this statistic to drill down to the [Blocking Lock Rate tab](#) of the Locks view.

Metrics

Immediately investigate a percentage much above zero so the situation does not mushroom.

You can easily drill down with Performance Center and discover the exact process(es) holding locks that are blocking out other user activity.

Troubleshooting

Once you discover a blocking lock situation, you can normally remedy it by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing, and usually results in other user processes completing in an instant. Performance Center makes discovering the blocked lock situation easier; preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in Sybase. You can change this behavior by modifying the Lock Wait Period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

Deadlock Rate

- [Metrics](#)
- [Troubleshooting](#)

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process' page or object. Each waits for the other to release the necessary lock. Sybase constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates the transaction. The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process' page or object. Each waits for the other to release the necessary lock. Sybase constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates it (the transaction). The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

The deadlock rate displays the percentage of times deadlocks occurred vs. the percentage of locks requested and immediately granted.

Metrics

You should immediately investigate a percentage much above zero to prevent the situation from mushrooming. You can easily drill down and discover the exact process(es) holding locks and deadlocks that are blocking out other user activity.

Troubleshooting

Well-designed applications can minimize deadlocks by always acquiring locks in the same order. You should always do updates to multiple tables in the same order.

Once Sybase discovers a deadlock, it takes action and remedies the situation. Performance Center makes it easier to discover how prevalent deadlock activity is on a system; preventing deadlocks from occurring in the first place is more difficult.

Those responsible for writing systems can minimize deadlocks by ensuring that applications acquire locks in the same order. Likewise, you should always do updates and other DML that act on multiple tables in the same order.

You can also shrink the amount of time that Sybase waits to check for deadlocks by modifying the deadlock checking period configuration parameter.

Device I/O Contention

- [Metrics](#)
- [Troubleshooting](#)

Device I/O Contention reflects the number of times a task or process was put to sleep while waiting for a semaphore for a particular database device.

When a task or process involves physical I/O, Sybase first fills out the block I/O structure and links it to a per engine I/O queue. If two or more Sybase engines request an I/O structure from the same device at the exact same time, the server puts one of them to sleep where it waits for the semaphore it needs.

Metrics

An I/O contention percentage consistently greater than 10% should be cause for concern.

Troubleshooting

Distributing the accessed objects across multiple devices can lessen contention for I/O device semaphores. You can also place devices on different physical drives to lessen contention at the operating system level as well.

Task Context Switches

- [Metrics](#)

Task Context Switches summarizes task-switch activity for all engines on SMP servers. You can use this statistic to observe the effect of reconfigurations.

Metrics

You might reconfigure a cache or add memory if tasks appear to block on cache search misses and to be switched out often. Then, check the data to see if tasks tend to be switched out more or less often.

Network Contention Rate

- [Metrics](#)
- [Troubleshooting](#)

The Sybase server normally sends and receives network packets at a regular rate. If the network begins to become saturated, Sybase could experience delays in network I/O. The network contention rate indicates the percentage of times network I/O activity was delayed.

Metrics

Seeing a network contention rate greater than 1% could indicate a challenged network structure.

Troubleshooting

Outside of ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase server.

I/O Vital Signs

The following I/O statistics are on the Sybase Home view:

- [Total Server Reads](#)
- [Total Server Writes](#)
- [Transaction Log Writes](#)
- [I/O Errors](#)

Total Server Reads

- [Metrics](#)

Total Server Reads reflect the total number of physical reads performed by the database server since the last refresh inside Performance Center.

Metrics

Large numbers of physical reads could reflect a too small data or procedure cache. You should examine the data and procedure cache hit rates to determine the overall effectiveness of logical vs. physical I/O.

Total Server Writes

The Total Server Writes value reflects total number of physical writes performed by the database server since the last refresh inside Performance Center.

Metrics

None.

Transaction Log Writes

Transaction Log Writes refers to the number of times Sybase wrote a transaction log page to disk since the last refresh inside Performance Center. When the current log page becomes full, Sybase writes it out to disk. Sybase also writes transaction log pages to disk after a transaction commits.

Metrics

None.

I/O Errors

- [Metrics](#)

The I/O Errors value reflects total number of I/O errors (errors during read and write operations) encountered by the database server since the last refresh inside Performance Center.

Metrics

You should investigate large numbers of I/O errors by examining the database error log.

Users Vital Signs

The following user statistics are on the Sybase Home view:

- [Total Connections](#)
- [Active Connections](#)
- [Committed Transactions](#)
- [Current Locks](#)

Total Connections

- [Metrics](#)
- [Troubleshooting](#)

The Total Connections statistic represents the total number of open threads, or connections, currently reported in the Sybase server. This number includes both active and inactive processes.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

You should view the total number of connections in light of the maximum number of processes allowed to connect to Sybase. The Number of User Connections parameter specifies the maximum number of user processes that can simultaneously connect to a Sybase server.

Troubleshooting

If the total number of connections approaches the number of user connections limit, you should:

- 1 Edit the configuration file for Sybase.
- 2 Increase the amount of number of user connections to a higher value.
- 3 Cycle the Sybase server when possible to allow the new value to take effect.

Active Connections

The Active Connections statistic represents the total number of active and open threads reported in the Sybase server. This number displays the number of processes actively performing work.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

None.

Committed Transactions

- [Metrics](#)

Committed Transactions reflects the number of transactions committed since the last refresh inside Performance Center. This includes transactions that meet explicit, implicit, and ANSI definitions for committed transactions. Note also that Sybase counts multidatabase transactions.

TIP: Click this statistic to drill down to the [Transactions tab](#) of the Users Detail view.

Metrics

Multidatabase transactions generally incur more overhead than single database transactions (for example, a transaction that modifies two databases is counted as two transactions). They usually involve more log activity, cause more ULC flushes, and involve two-phase commits between the different databases. You can improve performance by reducing the number of multidatabase transactions.

Current Locks

- [Metrics](#)
- [Troubleshooting](#)

The Current Locks statistic displays the total number of locks obtained/requested by processes in the database.

TIP: Click this statistic to drill down to the [All Locks tab](#) of the Locks view.

Metrics

The main thing to watch with respect to current locks is that no DML locks currently held on the system approach the number of locks limit specified in the Sybase server's configuration. The parameter number of locks limits how many locks can exist on the system at one time.

Troubleshooting

If the total number of locks approaches the number of locks limit, you should:

- 1 Ensure that user processes are efficiently using locks and are committing frequently to avoid excessive lock hold times before editing the current Sybase configuration.
- 2 Edit the configuration file for the Sybase server.
- 3 Increase the amount of number of locks to a higher value.
- 4 Cycle the Sybase server when possible to allow the new value to take effect.

Space Vital Signs

The following space statistics are on the Sybase Home view:

- [Databases Low on Space](#)
- [Logs Low on Space](#)

Databases Low on Space

- [Metrics](#)

- [Troubleshooting](#)

The Databases Low on Space statistic indicates if any databases in the server are approaching low levels of available free space. Although good to know, you need a more detailed listing by database to determine where any actual space shortages exist in the server. You can view this information in the Performance Center Space performance category view.

Metrics

If any one database begins to approach 90% used space, and is continuing to dynamically grow, you should take action to prevent any future space allocation errors.

Troubleshooting

If a database is approaching its limit on space, a DBA can either extend the database onto a new device, or if space exists on the device where the database currently resides, the DBA can allocate more space for the database on the current device.

Logs Low on Space

- [Metrics](#)
- [Troubleshooting](#)

This statistic indicates if any log for a database in the Sybase server is approaching low levels of available free space. Although good to know, a more detailed listing by database is needed to determine where any actual space shortages exist in the server. This information can be viewed in the Performance Center Space performance category view.

Metrics

If a database log's used space begins to approach 90%, you should take action to prevent any future space allocation errors. If the transaction log runs out of space, no transactions can take place in the database until you free up space in the log.

Troubleshooting

If a database log consistently approaches its limit on used space, there are a few actions a DBA can take to prevent a database from freezing.

A backup (dump) of the log can be taken:

- If log backups are not required for disaster recovery, the truncate log on checkpoint option can be set for the database. Setting this option deletes any space devoted to transactions in the log that have already completed when a database checkpoint occurs.
- You can extend the log onto a new device. Or, if space exists on the device on which the database log currently resides, you can allocate more space for the log on the current device.

Network Vital Signs

The following network statistics are on the Sybase Home view:

- [Network Requests](#)
- [Network Delays](#)
- [Bytes Received](#)
- [Bytes Sent](#)

Network Requests

The Network Requests statistic represents the total TDS packets received and sent since Performance Center was last refreshed.

Metrics

None.

Network Delays

- [Metrics](#)
- [Troubleshooting](#)

Network Delays represents the number of times I/O was delayed since Performance Center was last refreshed.

Metrics

Seeing a value much above zero could indicate a challenged network structure.

Troubleshooting

Outside of ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase server.

Bytes Received

This statistic represents the number of bytes received by Sybase since Performance Center was last refreshed.

Metrics

None.

Bytes Sent

Bytes Sent represents the number of bytes sent by Sybase to client applications since Performance Center was last refreshed.

Metrics

None.

Memory Statistics - Sybase

The Memory performance category view displays the following vital Sybase memory statistics:

- [Cache Allocations](#)
- [Dirty Buffers Grabbed](#)
- [Data Cache Activity](#)

- [Dirty Read Requests](#)
- [Dirty Read Restarts](#)
- [Large I/O Hit Rate](#)
- [Large I/O Acquired](#)
- [Large I/O Denied](#)
- [New Pages Allocated](#)
- [Procedure Cache Activity](#)
- [Procedure Reads from Disk](#)
- [Procedure Requests](#)
- [Procedure Removals](#)
- [Procedure Writes to Disk](#)
- [Session Leaders - Memory](#)

Data Cache Activity

- [Metrics](#)
- [Troubleshooting](#)

Data read from memory produces end-user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is something that the Sybase data cache tries to assist with.

The data cache hit activity rate is an excellent indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. The table below lists three key counters in Sybase used to arrive at this statistic:

Counter	Description
LOGICAL READS	Data read from memory for user requests.
PAGES PER I/O	The number of pages retrieved in a single I/O operation.
PHYSICAL READS	Data read physically from disk.

TIP: Click this category heading to drill down to the [Cache Activity tab](#) of the Memory Detail view.

Metrics

To help ensure excellent performance, you want to keep your cache hit rate in the neighborhood of 90% or greater. Lower amounts can be okay for user ad hoc databases where sporadic, large table scan operations occur. However, anything below this general threshold for normal databases can require tuning attention, and the adjustment of the Sybase memory tuning parameters.

If you are using named caches, you can drill down into the cache hit rates for each named cache. This helps you understand which objects/operations are depressing the overall cache hit rate for the server.

Troubleshooting

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or reduce the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when reducing the procedure cache, as this could reduce performance in the server as it relates to reading procedures in from disk.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the data cache (and any named caches), but in 12.5, if you want to reduce the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the data cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary as will an examination of unnecessary large table scan operations.

New Pages Allocated

The statistic reports the number of times that a new page was allocated in memory for Sybase since the last refresh in Performance Center.

Metrics

None.

Dirty Buffers Grabbed

- [Metrics](#)
- [Troubleshooting](#)

As information is requested from users, buffers are moved into and out of the Sybase data cache. Pages are also modified in the cache (termed dirty buffers) and need to be written out to disk. If Sybase has to wait for a dirty buffer to be written out to disk before a requested buffer is placed into the cache, performance can suffer.

This statistic represents the number of times Sybase found dirty buffers since the last refresh in Performance Center.

Metrics

Ideally, the dirty buffer grab statistic should stay close to zero.

Troubleshooting

Seeing high numbers for this statistic could indicate that the cache size is too small. You can want to look into carefully adjusting the total memory configuration parameter higher. However, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Dirty Read Requests

- [Metrics](#)

Sybase allows dirty reads, which are reads of uncommitted data. To accomplish a dirty read, Sybase uses a special lightweight protection mechanism to gain access to an object without using actual page locks. This statistic displays the number of dirty reads that occurred since the last refresh in Performance Center.

Metrics

Dirty read page requests can incur significant overhead if they are observed with many dirty read restarts.

Dirty Read Restarts

- [Metrics](#)
- [Troubleshooting](#)

Sybase allows dirty reads, which are reads of uncommitted data. To accomplish a dirty read, Sybase uses a special lightweight protection mechanism to gain access to an object without using actual page locks. A dirty read restart occurs when a dirty read is active on an object page, and another process makes changes to the page that cause the page to be deallocated in memory. The scan for the dirty read must be restarted. The amount shown for dirty read restarts are the number of restarts that occurred since the last Performance Center refresh.

Metrics

Values observed much above zero should serve as a signal that application modifications can be in order. Most applications should do everything possible to avoid restarts because of the large overhead they incur.

Troubleshooting

If the numbers observed for dirty read restarts are significant, you can want to look into modifying applications that use dirty reads to accomplish data acquisition.

ProcedureCache Activity

- [Metrics](#)
- [Troubleshooting](#)

The Sybase procedure cache is used to hold the definitions and query plans of stored procedures and triggers and is used for short-term memory needs like statistics and query plans needed for parallel queries. When a user executes a stored procedure, Sybase looks in the procedure cache for a query plan to use. If a query plan is available, Sybase places it on the most recently used (MRU) end of the memory chain and execution of the procedure begins. If no execution plan is in memory, or if all copies of the plan are currently being used, the query tree for the procedure is read in again from the data dictionary, optimized, put on the MRU end of the chain, and executed.

NOTE: Other operations, like CREATE INDEX, can also use the procedure cache even when no procedure is referenced.

The percentage of times that a procedure's plan and definition can be referenced in memory, the better the procedure execution time.

Metrics

A high procedure cache hit rate is a desirable thing. You should strive for a hit ratio between 95-100%, with 95% being a good performance benchmark for procedure code reference. Note that when a database is first started, the procedure cache hit rate is not at an optimal level because all code being used is relatively new, and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels, you should look into the possibility of increasing the amount of memory allocated to the cache.

NOTE: You can drill down into the procedure cache to view the procedures currently in memory along with how much memory they are consuming.

If there is not enough memory to load a requested procedure, or the maximum number of compiled objects is already in use, Sybase returns an error (normally a 701).

Troubleshooting

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase server, but in 12.5, if you want to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the procedure cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Procedure Requests

The Procedure Requests statistic reports the number of times that stored procedures were executed since Performance Center was last refreshed. Such a request could use either an unused copy of the procedure's query plan in memory or if no such copy exists, the procedure must be read in from disk.

Metrics

None.

Procedure Reads from Disk

- [Metrics](#)
- [Troubleshooting](#)

The Procedure Reads from Disk statistic reports the number of times since Performance Center was last refreshed that stored procedures were read from disk rather than copied in the procedure cache.

Metrics

You should examine this number in conjunction with the overall procedure cache hit rate. Observing large numbers in this statistic indicates a lower than ideal procedure cache hit rate. Note that when a database is first started, this statistic is likely larger than desired because all code being used is relatively new and as such, must be read in from disk and placed into the cache. If, however, after a solid hour or two of steady database time, the procedure cache hit rate has not increased to desirable levels and this statistic continues to sport high numbers, you should look into the possibility of increasing the amount of memory allocated to the cache.

Troubleshooting

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase server, but in 12.5, if you want to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the procedure cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Procedure Writes to Disk

This statistic reports the number of times since Performance Center was last refreshed that stored procedures were created.

Metrics

None.

Procedure Removals

- [Metrics](#)
- [Troubleshooting](#)

This statistic reports the number of times since Performance Center was last refreshed that stored procedures were aged out of the procedure cache.

Metrics

High numbers, along with a lower than desired procedure cache hit rate, could indicate too small a procedure cache.

Troubleshooting

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter and/or increase the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when increasing the procedure cache alone, as this could increase query response times due to more physical I/O being performed.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the Sybase server, but in 12.5, if you want to increase the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the procedure cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, another increase may be necessary. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Large I/O Hit Rate

- [Metrics](#)
- [Troubleshooting](#)

Large I/O can be enabled by splitting the default or any named cache into pools. By default, Sybase performs I/O operations based on a 2-KB page size. For queries where pages are stored and accessed in a sequential manner, it is possible to read many more data pages in a single I/O operation. Large I/O can greatly reduce disk access time when the right situations exist. Operations that routinely perform large table scans, access image or text data, do bulk copying, scan the leaf level of nonclustered indexes, or initiate DBCC tasks can benefit from large I/O.

If large I/O has been configured and is being used, you should observe a high percentage of hits (the number of times large I/O could be performed vs. the number of times large I/O requests were denied by the server). If large I/O is not configured, no large I/O activity should be present.

The table below describes the other large I/O statistics available on the Memory Detail view:

Statistic	Description
Large I/O Acquired	This statistic measures the number of times that a requested large I/O operation was performed since the last Performance Center refresh.
Large I/O Denied	This statistic measures the number of times since the last Performance Center refresh that large I/O could not be performed since the last Performance Center refresh. You should examine nonzero numbers in light of the overall large I/O hit rate. If large I/O is configured, but you observe nonzero values for Large I/O Denied and a low Large I/O hit rate is also present, you should configure more caches for large I/O use.

Metrics

As you might expect, if large I/O is in use, a high hit rate is desirable. You should strive for a hit ratio between 90-100%, with 90% being a good performance benchmark.

Troubleshooting

If large I/O is configured, but a low hit rate is being observed, you should configure more caches for large I/O use.

Session Leaders - Memory

- [Metrics](#)
- [Troubleshooting](#)

It is common for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Frequently, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system levels.

The leading memory session's display identifies the top users in the server with respect to memory consumption.

TIP: Click this category heading to drill down to the [Leading Sessions tab](#) of the I/O Detail view.

Metrics

If your server does not have an overabundance of memory, you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing.

Troubleshooting

You can use the Session Leaders - Memory statistic to find the users with the greatest current allocations of overall memory. Runaway processes can be immediately terminated from within the Performance Center Client.

Cache Allocations

The Sybase cache areas are devoted to facilitating the transfer of data and information between clients and the Sybase database. The table below describes their main tunable components:

Component	Description
Procedure Cache	Is used to hold the definitions and query plans of stored procedures and triggers and is used for short-term memory needs like statistics and query plans needed for parallel queries.
Data Cache	Maintains data blocks that are read from the database. Properly sizing the data cache (both default and named caches) goes a long way to improving response time performance.

Metrics

None.

Memory Detail View

The Memory Detail view includes the following tabbed pages:

- [Cache Activity](#)
- [Leading Sessions](#)
- [Page Activity](#)

Leading Sessions Tab

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. User connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system levels.

Performance Center displays information to help you find processes that are using the most memory on the server on the Leading Sessions tab of the Memory Detail view and the Memory tab of the Top Sessions view.

The table below describes the information available on these tabs:

Column	Description
PID	The process ID.
User Name	The logon name the session is using.
FID	The process ID of the worker process' parent.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The client machine name the session is using.
Program	The executable the process is using against the Sybase server.
Memory	The amount of memory currently allocated to the process.
Pct Mem Used	The percentage of memory currently used by the process.
Database	The database the process is attached to.
Command	The command the process is currently issuing.
Trans	The name of any active transaction.

NOTE: This information is available on both the Leading Sessions tab of the Memory Detail view and the [Memory tab](#) of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If your database server does not have an overabundance of memory, you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing.

Cache Activity Tab

- [Metrics](#)

Database administrators can configure multiple caches for the Sybase server. These can be used judiciously by a DBA to hold objects being requested for data and speed end use response times. The Cache Activity tab of the Memory Detail view contains information that you can use to see if configured caches are being used the way they were intended. The table below describes the information available on the Cache Activity tab of the Memory Detail view:

Column	Description
Cache Name	The name of the configured cache.
Hit Rate	The percentage of times that a data page was found in the particular cache vs. having to be read in from disk. Higher percentages are desirable because data read from memory can be accomplished in much less time than data read from disk.
Pct Used	The percentage of the cache currently being used.
Spinlocks	How many spinlocks the cache has experienced.
LRU Buffers	Indicates the number of buffers acting in accordance with the normal cache strategy of moving to the most recently used (MRU) end of the cache.
MRU Buffers	The number of buffers following a fetch-and-discard cache strategy.
Large I/O	The number of times large I/O requests were performed.
Dirty Reads	The number of dirty reads (reads of uncommitted data) performed.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Seeing low hit rates for any configured dynamic caches indicates that they are not large enough and that users experience slower than desirable response times. On the other hand, caches with zero hit rates can indicate that they are not being used and that their memory could be released to another cache or the operating system itself. Another indicator that this can be the case is a very low percent used reading.

Page Activity Tab

- [Metrics](#)

Pages are read into and deallocated from memory constantly in a dynamic Sybase environment. The Page Activity tab of the Memory Detail view displays how many pages were moved into and out of memory during a particular monitoring interval. The table below describes the information available on the Page Activity tab of the Memory Detail view:

Column	Description
Page Action	The type of paging action that occurred, either Pages Allocated or Pages Released. Allocated pages are ones that were moved into memory during the monitoring interval. Released pages are ones that were deallocated from memory during the monitoring interval.
Count	The number of pages moved.

Metrics

Seeing allocation and deallocation numbers that are close can indicate objects that have a fetch-and-discard cache strategy. If large table scans are continuously needed on a system, this can be a valid strategy (because large amounts of unnecessary data kept in a cache can crowd out needed data pages). If this is not the case, investigate further to ensure that objects have a valid cache strategy (most recently used or MRU) in place.

I/O Statistics - Sybase

The I/O performance category view displays the following vital Sybase I/O statistics:

- [Completed Disk I/Os](#)
- [Delayed Disk I/Os](#)
- [I/O Error Rate](#)
- [Outstanding Disk I/Os](#)
- [Requested Disk I/Os](#)
- [Server I/O Busy Rate](#)
- [Session Leaders - I/O](#)
- [Transaction Log Allocations](#)
- [Transaction Log Writes](#)

Requested Disk I/Os

- [Metrics](#)

This statistic reports the number of times that Sybase requested disk I/Os since Performance Center was last refreshed.

TIP: Click this statistic to drill down to the [Devices tab](#) of the I/O Detail view.

Metrics

The numbers for requested disk I/Os and completed I/Os should be very close. If there are large differences in these two values, it is likely that network contention or saturation is occurring.

Note that the value for requested I/Os includes every request that was initiated during the sample period. It is possible, however, that some of them completed after the sample period ended. These I/Os would then be excluded from the total number of completed I/Os, and can give the appearance that network problems exist when that is not the case. The number for completed I/Os can also be larger than the value for requested I/Os if I/O requests were made before Performance Center polled the server but were completed during the polling interval.

Completed Disk I/Os

- [Metrics](#)

The Completed Disk I/Os statistic reports the number of times that Sybase completed disk I/Os operations since Performance Center was last refreshed.

TIP: Click this statistic to drill down to the [Engines tab](#) of the I/O Detail view.

Metrics

The numbers for requested disk I/Os and completed I/Os should be very close. If there are large differences in these two values, it is likely that network contention or saturation is occurring.

Note that the value for requested I/Os includes every request that was initiated during the sample period. It is possible, however, that some of them completed after the sample period ended. These I/Os would then be excluded from the total number of completed I/Os, and can give the appearance that network problems exist when that is not the case. The number for completed I/Os can also be larger than the value for requested I/Os if I/O requests were made before Performance Center polled the server but completed during the polling interval.

Outstanding Disk I/Os

- [Metrics](#)

The Outstanding Disk I/Os statistic reflects the maximum number of I/Os pending for Sybase since the last refresh in Performance Center.

TIP: Click this statistic to drill down to the [Engines tab](#) of the I/O Detail view.

Metrics

If nonzero numbers are observed for both outstanding and delayed disk I/Os, there could be a problem in many areas. The table below describes these areas:

Area	Description
Delayed Disk I/O	This statistic indicates the number of I/Os delayed for Sybase since the last refresh in Performance Center.
Transaction Log Writes	This statistic indicates the number of times Sybase wrote a transaction log page to disk since the last refresh in Performance Center. When the current log page becomes full, it is written to disk. Transaction log pages are also written to disk after a transaction commits.
Transaction Log Allocations	This statistic indicates the number of times since Performance Center was last refreshed that additional pages were allocated to the transaction log. You can use this statistic for getting a feel for the rate of transaction log growth.
Server I/O Busy Rate	This statistic indicates number of seconds in CPU time that Sybase has spent doing input and output operations. Seeing a consistent rate of 90-100% could indicate a CPU bound server.

Transaction Log Writes

Transaction Log Writes refers to the number of times Sybase wrote a transaction log page to disk since the last refresh inside Performance Center. When the current log page becomes full, Sybase writes it out to disk. Sybase also writes transaction log pages to disk after a transaction commits.

Metrics

None.

Transaction Log Allocations

Transaction Log Allocations refers to the number of times since Performance Center was last refreshed that additional pages were allocated to the transaction log. This statistic gives you a feel for the rate of transaction log growth.

Metrics

None.

Session Leaders - I/O

- [Metrics](#)

Heavy I/O activity in a system can indicate that the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are responsible for 75% or more of the I/O activity. It may be that your system is running a large batch load or other typical processes, which are perfectly okay. On the other hand, it may be a runaway process or other rogue connection, which you need to track down and possibly eliminate.

The leading I/O session's display lets you see who the leading sessions are in your system with respect to I/O.

NOTE: This statistic displays on both the [Users performance category view](#) and the I/O performance category view.

TIP: Click this statistic to drill down to the [Leading Sessions tab](#) of the I/O Detail view.

Metrics

One or two users consuming more than 75% of the total I/O load can indicate a runaway or improper process. You can drill into the I/O activity of all users and quickly see if this is the case.

I/O Error Rate

- [Metrics](#)

The I/O Error Rate statistic represent the percentage of times I/O errors occurred in the server. I/O Error Rate is a percentage based on Total I/O (the sum the physical reads and writes.)

Metrics

You should investigate large numbers of I/O errors by examining the error log.

- [Delayed Disk I/Os](#)
- [Total I/O](#)

Delayed Disk IOs

- [Metrics](#)
- [Troubleshooting](#)

The Delayed Disk IOs value is the number of I/Os delayed when the limit on disk I/O structures is reached.

Metrics

When Adaptive Server exceeds the number of available disk I/O control blocks, I/O is delayed because Adaptive Server requires that tasks get a disk I/O control block before initiating an I/O request.

Troubleshooting

If you see a non-zero value for delayed disk I/Os, try to add to the number of available disk I/O control blocks by increasing the configuration parameter.

Total I/O

- [Metrics](#)

The Total I/O statistic represents the total number of physical reads and writes.

Metrics

None.

I/O Errors

- [Metrics](#)
- [Troubleshooting](#)

I/O error rate reflects total number of I/O errors (errors during read and write operations) encountered by the server since the last refresh inside Performance Center. The I/O errors rate is a percentage based on Total I/O (the sum of the physical reads and writes).

Metrics

You should observe few, if any errors.

Troubleshooting

If you notice any errors, you should check the Sybase error log for details.

I/O Busy

- [Metrics](#)
- [Troubleshooting](#)

The I/O Busy statistic represents the number of clock ticks in the sample interval during which the user task performed I/O operations.

Metrics

High numbers indicate an I/O-intensive process. If idle time is also high, the application could be I/O bound.

Troubleshooting

The application might achieve better throughput if you assign it a higher priority, bind it to a lightly loaded engine or engine group, or partition the application's data onto multiple devices.

- [Average Number of Writes per Log Page](#)

Average Number of Writes per Log Page

- [Metrics](#)

The Average Number of Writes is the average number of times each log page was written to disk.

Metrics

In high throughput applications, this number should be as low as possible. If the transaction log uses 2K I/O, the lowest possible value is 1; with 4K log I/O, the lowest possible value is .5, since one log I/O can write 2 log pages.

In low throughput applications, the number will be significantly higher. In very low throughput environments, it may be as high as one write per completed transaction.

The following statistics are used on the Performance Center for Sybase I/O page to succinctly communicate the general overall performance levels of I/O:

- [Top I/O Hogs](#)

Top I/O Hogs

- [Metrics](#)
- [Troubleshooting](#)

The Top I/O Process statistic identifies the processes that has currently caused the most I/O usage on the database.

The table below describes the information available from the Top I/O Hogs view of the I/O page:

Column	Description
SPID	The process ID.
Login	The login name the session is using.
Physical I/O	The physical amount of I/O the process is using.
% Used	Percent of total I/O on the server this process is consuming.

Metrics

None.

Troubleshooting

If any one session uses more than 50% of a total resource (CPU, memory, etc.) you should drill down into that particular session and investigate the cause.

I/O Detail View

The following tabbed pages are available on the I/O Detail view:

- [Devices](#)
- [Engines](#)
- [Index Scans](#)
- [Leading Sessions](#)

Leading Sessions Tab

- [Metrics](#)

When a system undergoes heavy I/O activity, sometimes you find that all the user connections are contributing somewhat equally to the overall load. Frequently, however, one or two user connections are responsible for 75% or more of the I/O activity. It may be that a large batch load or other typical process is running that is perfectly okay for your system. Alternatively, it may be a runaway process or other rogue connection that you should track down and eliminate.

Performance Center displays information to help you find processes that are using the most memory on the server on the Leading Sessions tab of the I/O Detail view and the I/O tab of the Top Sessions view.

The table below describes the information available on these tabs:

Column	Description
PID	The process ID.
User Name	The logon name the session is using.
FID	The process ID of the worker process' parent.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The client machine name the session is using.
Program	The executable the process is using against the Sybase server.
Physical I/O	The amount of I/O the process has currently accumulated.
Pct I/O Used	The percentage of overall I/O that can be attributed to the process.
Database	The database to which the process is attached.
Command	The command the process is currently issuing.
Transaction	The name of any transaction.
Blocked	Indicates if the process is currently blocked by another process.
Time Blocked	The amount of time that the process has been blocked, in seconds.

NOTE: This information is available on both the Leading Sessions tab of the I/O Detail view and the [I/O tab](#) of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Pinpointing sessions with abnormally high I/O activity relative to other sessions in the system helps you identify accounts dragging down overall system performance. You should examine the activity of each session to determine the system workload and to determine if you can reduce the workload or tune the system for better performance.

Devices Tab

- [Metrics](#)
- [Troubleshooting](#)

Devices are accessed repeatedly in a dynamic Sybase environment to satisfy end user requests for data, to handle write activity that records transactions in a database's transaction log, and to manage other I/O operations. Viewing the I/O activity for each device is a good way to see what the "hot" devices are in a Sybase server with respect to I/O usage. The same information can be used to spot heavy database usage in systems where the device-to-database mappings are one-to-one.

The Devices tab of the I/O Detail view displays information that is useful in determining device I/O patterns. The table below describes the information available on the Devices tab of the I/O Detail view:

Column	Description
Device Name	The name of the device.
Physical Name	The name used by the operating system to identify the device.
Hit Rate	The rate at which accesses to a device were granted immediately compared to the total number of requests.
Reads	The total number of read operations for the device.
Writes	The total number of write operations for the device.
Total I/O	The combined total of read and write operations.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If devices have one-to-one relationships with user databases, you can quickly tell which databases are experiencing the most demand on a server. If the device-to-physical drive/file system is a one-to-one relationship, you can also spot which physical drives on a server are getting the heaviest workouts. For servers that have many drives, it is normally desirable to spread devices across different physical drives and controllers so contention does not occur at the disk level. In addition, separating databases and their corresponding logs is normally recommended so that each is located on a distinct physical drive. If possible, write-intensive devices, like log devices, are best suited for non-RAID5 scenarios.

Troubleshooting

If device hit rates are low, you can add more devices or redistribute objects among different devices. Typically, segments can be used to redistribute objects among different devices or physical hard disks. Common techniques include placing tables and indexes on different segments and partitioning large tables.

If device loads appear skewed (one device has much more activity than others), you should focus attention on that device. Again, redistributing objects can lessen the device's workload.

Engines Tab

- [Metrics](#)
- [Troubleshooting](#)

In symmetric multiprocessing (SMP) environments, a DBA can configure the Sybase server to use more than one "engine," which represents a certain amount of CPU power. By default, Sybase configures one engine for use. If you have a server machine with multiple CPUs, you can enable more engines to take advantage of the machine's additional processing ability.

The Engines tab of the I/O Detail view displays information with respect to how each engine is handling I/O. The table below describes the information available on the Engines tab of the I/O Detail view:

Column	Description
Engine	The name of the configured engine.
Completed I/Os	The number of I/Os completed during the sample interval.
Outstanding I/Os	The number of I/Os left outstanding during the sample interval.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If the number of outstanding I/Os remains high or increases during periods of heavy activity, there may not be enough engines configured for the system.

Troubleshooting

If the server machine has multiple CPUs, you can configure more engines for Sybase to use by following this process:

- 1 Use the sp_configure procedure to change the current engine configuration. For example, to change the number of engines from one to two, you would run: 'sp_configure "max online engines",2'
- 2 Stop and restart the Sybase server.

Index Scans Tab

- [Metrics](#)

Indexes are accessed frequently in dynamic Sybase server environments. The type of index access often determines the response time an end user experiences. Single row index accesses are the quickest, and complete index scans are the most time consuming (for large indexes at least).

The Index Scans tab of the I/O Detail view presents information with respect to index scans. The table below describes the information available on the Index Scans tab of the I/O Detail view:

Column	Description
Scan Type	The type of index scan.
Count	The number of scans per type for the sample interval.

Metrics

There are two basic scan types - ascending and descending. For ascending scans, Sybase moves up the index page chain from beginning to end. Sybase follows the page chain in reverse order for descending scans. Descending scans are normally the result of requests made for data by descending column order.

Within ascending and descending scans, a data-only lock, or DOL, styled access can also occur.

Space Statistics - Sybase

The Space performance category view displays the following vital Sybase space statistics:

- [Database Overview](#)
- [Log Overview](#)
- [Device Overview](#)

Database Overview

- [Metrics](#)
- [Troubleshooting](#)

A Sybase server contains many databases, some of which are devoted to system-level activities (the master and tempdb databases, for example) and others that hold user data. The database overview displays details about the space situation for each database in a Sybase server, including the total, used, and free space. The percentage used amount for each database is also shown.

TIP: Click this statistic to drill down to the [Object Detail tab](#) of the Space Detail view.

Metrics

If a database's used space percent amount goes above 90%, and the database is dynamic in nature (meaning that users are constantly adding and modifying data), then you should take action to ensure that the database does not run out of available free space.

Troubleshooting

If the percent used amount of a database is approaching problematic levels, there are two ways you can rectify the situation:

- 1 If the database device that the database currently resides on contains additional free space, you can ALTER the database to consume more available space on that device.
- 2 If the database device that the database currently resides on does not contain additional free space, you can do one of the following:
 - Create a new device and issue an ALTER for the database to use that device for space in addition to those currently used.
 - Choose another existing device that has free space and ALTER the database to use that device in addition to those currently in use.

Log Overview

- [Metrics](#)
- [Troubleshooting](#)

Each database in Sybase has a transaction log, which can be placed on the same device as its database or assigned to a separate device for backup and performance purposes. The log overview displays details concerning each transaction log in the Sybase server and includes the total, used, and free space amounts. The percent used amount for each log is also shown.

Metrics

If any log's used space exceeds 90%, you should take action to ensure that the log does not run out of available free space.

Troubleshooting

There are many things a DBA can do to ensure that a database's log does not run out of available free space:

- 1 First, most transactional-oriented databases should have their logs assigned to a device separate from the database. Reasons for doing so include:
 - It lets you have more granular and point-in-time backups because a log separate from its database's device can make use of the dump transaction command.
 - It prevents competition for space between the log and the database itself.
 - It allows the log to be monitored for space more effectively.
 - It improves performance.
 - It enables better recovery from hard disk crashes because you can dump your transaction log, even when your data device is on a server's damaged hard disk.
- 2 If the database is not critical in nature, you can set the truncate log on checkpoint option (trunc log on chkpt), which eliminates any inactive space in the log when a database checkpoint occurs.
- 3 Critical databases needing higher levels of recovery should have schedules established that regularly perform transaction log dumps. Doing so ensures better recovery scenarios as well as a reduced risk of the transaction log running out of space.
- 4 If a critical transaction log becomes full, it can be impossible to use standard procedures to dump transactions and reclaim space. The dump operation incorporates the no log or truncate only options.
- 5 If a transaction log continuously approaches dangerously low levels of free space, you should expand the database (if the transaction log is on the same device as the database) or expand the log onto its current device (if room exists) or a new device.

You should also be on the lookout for large load or data modification operations that do not make use of prudently timed commit points. A single, large transaction has the ability to overwhelm any transaction log because only inactive space in the transaction log is removed from log dumps or truncation operations.

Device Overview

- [Metrics](#)
- [Troubleshooting](#)

Database devices are the logical containers for databases and transaction logs. Each device is mapped to a physical file that is located on a database server's hard drive. The device overview displays details about every defined device on a Sybase server, including the total, used, and free space amounts. The percent of used space is also displayed.

TIP: Click this statistic to drill down to the [Device Detail tab](#) of the Space Detail view.

Metrics

The environment of the particular Sybase server as well as your work style dictate the metrics to use in evaluating a device that is running into trouble with space. Many DBAs create devices that parallel a single, corresponding database in size, and therefore, such devices show 100% utilization. Other DBAs create large devices that are not completely utilized.

Troubleshooting

If a device has become too full, a DBA can begin the process of manually relocating databases from it onto other devices. The process of moving logs is somewhat easier and can be accomplished via singular commands (sp_logdevice).

Space Detail View

The following tabbed pages are available on the Space Detail view:

- [Cache Detail](#)
- [Device Detail](#)
- [Object Detail](#)
- [Segment Detail](#)

Device Detail Tab

- [Metrics](#)
- [Troubleshooting](#)

The Device Detail tab of the Space Detail view presents details about a selected device located on the monitored Sybase server. The table below describes the information available on the Device Detail tab of the Space Detail view:

Column	Description
Database Name	The name of a database that uses all or part of the device for its database space, transaction log space, or both.
Space	The amount of space used by the database on the device.
Space Type	The type of space allocated (database, log, or both).

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

The environment of the particular Sybase server as well as the DBA's work style dictate the metrics to use in evaluating a device that is running into trouble with space. Many DBAs create devices that parallels a single, corresponding database in size, and therefore, such devices always show 100% utilization. Other DBAs create large devices that are not completely utilized.

Troubleshooting

If a device has become too full, a DBA can begin the process of manually relocating databases from it onto other devices. The process of moving logs is somewhat easier and can be accomplished via singular commands (sp_logdevice).

Cache Detail Tab

- [Metrics](#)
- [Troubleshooting](#)

Caches retain frequently requested data and object information in memory. They can potentially boost performance by allowing users to read or write data in memory instead of to and from physical disk. Information requests satisfied in memory equate to much better response times than if that same data requires physical I/O access.

Named data caches can help achieve substantially greater concurrency and increase the likelihood that Sybase reads data from memory and not disk by letting you bind particularly active database objects to dedicated data caches. In addition, you can boost performance by creating memory pools, each configured for different I/O throughput (2 KB-16 KB), within every named data cache.

The table below describes the information available on the Cache Detail tab of the Space Detail view:

Column	Description
Cache Name	The name of the data cache.
Status	The status of the cache (Active, Activation Pending, Deletion Pending).
Mode	Mixed or log only.
Configured Size	The defined size of the cache.
Running Size	The actual running size of the cache.
Overhead	The amount of overhead required to manage the cache.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

The data cache hit rate is an excellent indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. To help ensure maximum performance, you want to keep your cache hit rate in the neighborhood of 90% or greater. Lower amounts can be okay for user ad hoc databases where sporadic, large table scan operations occur. But anything below this general threshold for normal databases can require tuning attention and the adjusting the Sybase memory tuning parameters.

If you are using named caches, you can drill down into the cache hit rates for each named cache (from the Memory performance category view). This helps you understand which objects/operations are depressing the overall cache hit rate for the server.

Troubleshooting

If a problem is found in Sybase servers, versions 11-12, you can increase the amount of the total memory configuration parameter or reduce the percentage of memory allocated to the procedure cache (by default, the data cache assumes any free memory left over after Sybase has met its kernel and procedure cache needs). Take care when reducing the procedure cache, as this could reduce performance in the server as it relates to reading procedures in from disk.

For Sybase 12.5, the total memory configuration parameter can again be increased to provide more memory for the data cache (and any named caches), but in 12.5, if you want to reduce the size of the procedure cache, note that it is now configured in terms of literal size instead of a percentage of the overall configured memory.

Once the data cache has been adjusted, monitor Sybase to see if the cache hit rate improves. If it does not, then another increase may be necessary as will an examination of unnecessary large table scan operations. Also, keep a careful eye on the actual machine's memory limits and swap activity. Increased swap activity can be indicative of too little memory left for the server machine.

Segment Detail Tab

- [Metrics](#)
- [Troubleshooting](#)

Segments are instruments used for mapping objects (tables and indexes) to specific physical locations on disk. Mapping certain objects to defined segments allows you to potentially increase I/O throughput by placing heavily used tables and indexes on different physical devices. You can place tables and indexes on segments by including placement statements in your CREATE TABLE or CREATE INDEX statements.

The Segment Detail tab of the Space Detail view lets you view information about segments for selected databases. The table below describes the information available on the Segment Detail tab of the Space Detail view:

Column	Description
Segment Name	The name of the segment.
Total (KB)	Represents the total amount of space defined for the segment, in KB. Typically this equates to the amount of space on the device where the segment resides.
Used (KB)	Represents the amount of space used by the segment alone, in KB. Segments containing no objects will show zero space used.

NOTE: This information is available on both the Segment Detail tab of the Space Detail view and the Segment Detail grid on the [Storage tab](#) of the Database Detail view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

It is often good practice to separate tables and their corresponding indexes onto different physical devices to minimize I/O contention at the server level. Segments can be used to accomplish this within Sybase. If you have hot tables and indexes within your system, investigate the use of segments to place such objects on separate, faster physical devices.

Troubleshooting

Hot objects can be relocated onto different devices through the use of the sp_placeobject procedure.

Object Detail Tab

Tables and indexes comprise every Sybase database. The Object Detail tab of the Space Detail view presents space-related information about tables and indexes for a selected database. The table below describes the information available on the Object Detail tab of the Space Detail view:

Column	Description
Owner	The owner of the object.
Object Name	The name of the table or index.
Type	The type of object (table or index).
Rows	The number of rows currently in the table.
Reserved (KB)	The amount of space reserved for the object, in KB.
Data (KB)	The amount of space consumed by data, in KB.
Indexes (KB)	The amount of space used by an index, in KB.
Unused (KB)	The amount of unused space (free space) that the object contains, in KB.
Segment Name	The segment name that the object is assigned to.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Databases Statistics - Sybase

- [Suspect Databases](#)

Databases with Suspect Pages

- [Metrics](#)
- [Troubleshooting](#)

Suspect objects normally indicate an internal problem in a server. They can also indicate that physical damage has occurred to part of a database.

Metrics

Suspect objects have no place in a production database. If you identify any suspect objects, you should take immediate action.

Troubleshooting

If the suspect object is an index, you could try dropping and recreating it, or use the DBCC REINDEX command. Other damaged objects can complete rebuilding the database. A suspect database can be a difficult thing to recover from quickly. There are times when the cache of the database is suspect and not the database itself. Stopping and starting the Sybase server can verify if this is the case. If the database itself is actually damaged, there could be a true recovery situation. The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You would then need to recreate the database and perform a recovery operation using the most recent database dump.

Suspect Databases

- [Metrics](#)
- [Troubleshooting](#)

The Suspect Databases value represents the number of suspect databases on the server.

Metrics

None.

Troubleshooting

It can be difficult to recover from a suspect database quickly. There are times when the cache of the database is suspect and not the actual database itself. Stopping and starting the Sybase server can verify if this is the case. If the database itself is actually damaged, there could be a true recovery situation. The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You would then need to recreate the database and perform a recovery operation using the most recent database dump.

The Databases performance category view displays Sybase [Database Summary](#) statistics.

Database Summary

- [Metrics](#)
- [Troubleshooting](#)

The Database Summary display lists every database in a Sybase server and includes an overview of the activity currently being experienced in each database. It also shows demographic information about each database, which includes the database owner, the last transaction dump date, and if the database itself or any object contained within it is suspect.

TIP: Click this statistic to drill down to the [Suspect Objects tab](#) of the Database Detail view.

Metrics

Databases with high numbers of blocked users could signal a contention problem. You should immediately investigate suspect databases or database objects.

Troubleshooting

The action you need to take depends on the situation you are watching. The table below describes four situations that you can recognize and fix:

Situation	Description
Blocked Users	You should investigate lock contention problems in detail. If the blocking lock situation has been long in duration, or involves many users, you should consider terminating the process holding the offending locks.
Suspect Objects	If Performance Center has indicated that a database contains suspect objects, the first thing to do is drill down to find exactly which objects are suspect. You can also do this manually by running a DBCC against the database. If the object is an index, you could try dropping and recreating it, or use the DBCC REINDEX command.
Suspect Database	A suspect database can be a difficult thing from which to quickly recover. There are times when the actual cache of the database is suspect and not the actual database itself. Stopping and starting the Sybase server can verify if this is the case. If the database itself is actually damaged, there could be a true recovery situation. The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You would then need to recreate the database and perform a recovery operation using the most recent database dump.
Suspect Pages	If you have to drop a table or index containing suspect pages, you must use a transaction in the master database. If you don't, the drop fails because it needs to delete entries for the suspect pages from the master database.

Database Detail View

The following tabbed pages are available on the Database Detail view:

- [Index Statistics](#)
- [Options](#)
- [Storage](#)
- [Suspect Objects](#)
- [Table Statistics](#)

Suspect Objects Tab

- [Metrics](#)
- [Troubleshooting](#)

Suspect objects normally indicate an internal problem in a server or that physical damage has occurred to part of a database. The Suspect Objects tab of the Database Detail view presents objects in a Sybase server that have been marked suspect for a selected database. The table below describes the information available on the Suspect Objects tab of the Database Detail view:

Column	Description
Owner	The owner of the suspect object.
Table Name	The name of the table.
Index Name	The name of any suspect index.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Suspect objects have no place in a production database. If any suspect objects are identified, you should immediately take action.

Troubleshooting

If the suspect object is an index, you could try dropping and recreating it, or use the DBCC REINDEX command. Other damaged objects can complete rebuilding the database. A suspect database can be a difficult thing to recover from quickly. There are times when the cache of the database is suspect and not the database itself. Stopping and starting the Sybase server can verify if this is the case. If the database itself is actually damaged, there could be a true recovery situation. The suspect database can be dropped using the DBCC DBREPAIR DROPDB command. You would then need to re-create the database and perform a recovery operation using the most recent database dump.

Table Statistics Tab

- [Metrics](#)
- [Troubleshooting](#)

Understanding the placement, shape, and organization of your tables enhances your ability to make correct decisions regarding their use and maintenance. The Table Statistics tab of the Database Detail view communicates useful measures to gauge the overall condition of the tables in a selected database. The [first grid](#) displays table demographic information. The [second grid](#) displays cache binding information. The table below describes the information available in the Table Demographics grid on the Table Statistics tab of the Database Detail view:

Column	Description
Owner	The owner of the table.
Table Name	The name of the table.
Segment Name	The segment that holds the table.
Last Statistic Date	The date and time when statistics were last gathered for the table.
Row Count	The reported number of rows in the table.
Reserved	The amount of space reserved (in KB) of the table.
Avg Row Size	The average size of a row in the table.

The table below describes the information available in the Cache Bindings grid on the Table Statistics tab of the Database Detail view:

Column	Description
Owner	The owner of the table.
Table Name	The name of the table.
Cache Name	The cache to which the table has been bound.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

There are a few data points worth noting with regard to the information presented in the Table Statistics tab of the Database Detail view:

- Heavily used tables and indexes can benefit from being separated onto different segments that correspond to different physical hard drives.
- The Sybase cost-based optimizer may not be choosing the proper access path for tables with out-of-date, or without any, statistics. Keeping dynamic objects current with respect to their internal statistics helps ensure that the Sybase kernel makes the proper decisions on how to satisfy user requests for data.
- Frequently accessed small - or look up - tables can benefit from being bound to a special cache that is used to retain their often-accessed data in memory.

Troubleshooting

To change the segment a table or index belongs to, you can use the `sp_placeobject` procedure. Keeping table statistics current can be handled by judicious use of the `UPDATE STATISTICS` command. Binding frequently accessed tables can be accomplished via the `sp_bindcache` procedure.

Index Statistics Tab

- [Metrics](#)
- [Troubleshooting](#)

Understanding the placement, shape, and organization of your table's indexes enhances your ability to make the right decisions about using and maintaining them. The Index Statistics tab of the Database Detail view communicates a number of useful measures to gauge the overall condition of the indexes in a selected database. The [first grid](#) displays index demographics. The [second grid](#) displays cache bindings. The table below describes the information available in the Index Demographics grid on the Index Statistics tab of the Database Detail view:

Column	Description
Owner	The owner of the table.
Index Name	The name of the index in <table name>.<index name> fashion.
Segment Name	The segment that holds the table.
Reserved (KB)	The amount of space reserved of the table, in KB.
Tree Depth	The height of the index. Can hold bogus values for indexes where statistics have not been gathered.
Leaf Count	Number of leaf pages in the index.
Leaf Row Size	The average size of a leaf row in the index.

The table below describes the information available in the Cache Bindings grid on the Index Statistics tab of the Database Detail view:

Column	Description
Owner	The owner of the table.
Table Name	The name of the table.
Index Name	The name of the index.
Cache Name	The cache to which the table has been bound.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

There are a few points worth noting with regard to the information presented in the Index Statistics view:

- Heavily used tables and indexes can benefit from being separated onto different segments that correspond to different physical hard drives.
- The Sybase cost-based optimizer could not be choosing the proper access path for tables that have indexes with out-of-date, or without any, statistics. Keeping dynamic objects current with respect to their internal statistics helps ensure that the Sybase kernel makes the proper decisions on how to satisfy user requests for data.
- Indexes that are frequently scanned can benefit from being bound to a special cache that can be used to retain their often-accessed data in memory.

Troubleshooting

To change the segment a table or index belongs to, you can use the `sp_placeobject` procedure. Keeping index statistics current can be handled by judicious use of the `UPDATE STATISTICS` command. Binding frequently scanned indexes can be accomplished via the `sp_bindcache` procedure.

Storage Tab

A database can contain a number of segments you can use to physically separate tables and indexes onto different physical devices. A database can also be bound to a specific data cache in hopes of improving performance through more frequent memory access. The [first grid](#) displays segment use by a database. The [second grid](#) displays cache use by a database.

The table below describes the information available in the Segment Detail grid on the Storage tab of the Database Detail view:

Column	Description
Segment Name	The name of the database segment.
Total (KB)	The total amount of space allocated to the segment, in KB. Note that this normally corresponds to the device allocations for a database.
Used (KB)	The amount of space used by the segment to hold tables and indexes, in KB. Note that space used by other segments that make use of the same device are not included in the total.

NOTE: The information in the Segment Detail grid is also available on the [Segment Detail tab](#) of the Space Detail view.

The table below describes the information available in the Cache Bindings grid on the Storage tab of the Database Detail view:

Column	Description
Cache Name	The name of the data cache.
Status	The status of the cache (Active, Activation Pending, Deletion Pending).
Configured Size (MB)	The defined size of the cache, in MB.
Running Size (MB)	The actual running size of the cache, in MB.
Overhead (MB)	The amount of overhead required to manage the cache, in MB.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

None.

Options Tab

- [Metrics](#)

A database can have a number of options set to determine behavior like transaction log usage, database access, etc. The Options tab of the Database Detail view presents all available database options for a selected database along with which options are in use. The table below describes the information available on the Options tab of the Database Detail view:

Column	Description
Option	The name of the database option.
Setting	Indicates whether the option is in use.

Metrics

Depending on how it is used, a database can have many or few options set. For example, a production database that needs near point-in-time recovery abilities would not have the truncate log on checkpoint option set, because doing so would negate the use of transaction log dumps. However a development database that is loaded/reloaded many times a day with test data would likely make use of the option to prevent the transaction log from filling up. You should examine the specific needs of each database under your care to determine which options are right for it.

Contention Statistics - Sybase

The Contention performance category view displays the following vital Sybase contention statistics:

- [Logical Lock Contention](#)
- [Address Lock Contention](#)
- [Group Commit Sleeps](#)
- [Modify Conflicts](#)
- [Device I/O Contention](#)
- [Disk I/O Structures](#)

- [Server Configuration Limit](#)
- [Engine Configuration Limit](#)
- [Operating System Limit](#)
- [Lock Contention](#)
- [Blocked](#)
- [Current Locks](#)
- [Deadlock Rate](#)
- [Deadlocks](#)
- [Network Contention Rate](#)
- [Network Requests](#)

Logical Lock Contention

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all processes on a small system, and can cause major headaches on large systems. Although Sybase supports flexible locking mechanisms, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

The Logical Lock Contention statistic reflects the number of times a task/process was switched out because of contention over table/page locks.

Metrics

Seeing consistent high numbers for this statistic should result in you investigating the lock details of your server before the situation has a chance to mushroom.

You can easily drill down and discover the exact process(es) holding locks that are blocking out other user activity.

Troubleshooting

Once you discover a blocking lock situation, you can normally remedy it by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing, and usually results in other user processes completing in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is where it gets tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do. You can also check to see if your queries are doing deferred and direct expensive updates, which can cause additional index locks.

By default, all processes wait indefinitely for locks in Sybase. You can change this behavior by modifying the lock wait period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

Address Lock Contention

This statistic reflects the number of times a task/process was switched out because of memory address lock problems. Sybase acquires address locks during update operations, and for index pages, OAM pages, allocation pages, and occasionally on data pages when page splits are performed.

Metrics

None.

Group Commit Sleeps

- [Metrics](#)
- [Troubleshooting](#)

For databases needing high throughput, a large log I/O size is very important to prevent disk queuing problems on the transaction log. Group commit sleeps reflect the number of times a task performed a transactional commit operation and was put to sleep by the server until data in the log was written to disk.

Metrics

You should examine group commit sleeps in conjunction with the number of committed transactions (found on the Users performance category view).

Troubleshooting

A high count for group commit sleeps is not necessarily a problem if the server's transaction rate is low. If there are a significant number of transactions that result in group commit sleeps, and the log I/O size is greater than 2 KB, a smaller log I/O size can help to reduce commit time by causing more frequent page flushes.

Other factors that can effect group commit sleeps are the size of the server run queue and the speed of the hard disk where the log is located.

Modify Conflicts

For some database operations, Sybase uses a specialized protection mechanism to get exclusive access to a page without using actual restrictive page locks. Examples include accessing certain system tables and dirty reads on data pages. These actions need exclusive access to the page in question, even though they do not actually modify it. Modify conflicts record how many times these actions conflicted with other processes needing true database locks.

Metrics

None.

Device I/O Contention

- [Metrics](#)
- [Troubleshooting](#)

Device I/O Contention reflects the number of times a task or process was put to sleep while waiting for a semaphore for a particular database device.

When a task or process involves physical I/O, Sybase first fills out the block I/O structure and links it to a per engine I/O queue. If two or more Sybase engines request an I/O structure from the same device at the exact same time, the server puts one of them to sleep where it waits for the semaphore it needs.

Metrics

An I/O contention percentage consistently greater than 10% should be cause for concern.

Troubleshooting

Distributing the accessed objects across multiple devices can lessen contention for I/O device semaphores. You can also place devices on different physical drives to lessen contention at the operating system level as well.

Disk I/O Structures

- [Metrics](#)
- [Troubleshooting](#)

The user community can become very dissatisfied when a Sybase server begins to experience I/O delays. When such I/O delays occur, you should investigate the various Sybase or operating system limits. It could be that I/O operations are being blocked by one or both.

The Disk I/O Structures statistic represents the number of I/O delays caused by Sybase when it reaches the disk I/O structures limit. When Sybase exceeds the number of available disk I/O control blocks, I/O is deferred. This is because Sybase requires that any task get a disk I/O control block before it begins an I/O request.

Metrics

You should be concerned if you consistently see numbers above zero.

Troubleshooting

If you continue to see nonzero numbers for this statistic, you can try increasing the number of available disk I/O control blocks. Do this by increasing the configuration parameter disk I/O structures.

Server Configuration Limit

- [Metrics](#)
- [Troubleshooting](#)

When a Sybase server begins to experience I/O delays, the result can be a very dissatisfied user community. When such problems begin to occur, things to investigate include the various Sybase or operating system limits. It could be that I/O operations are being blocked by one or both.

The Server Configuration Limit statistic shows nonzero numbers if Sybase has exceeded its limit for the number of asynchronous disk I/O requests that can be outstanding for the entire server at one time.

Metrics

You should be concerned if you consistently see numbers above zero.

Troubleshooting

If you continue to see nonzero numbers for this statistic, you can raise this limit using sp_configure with the max async I/Os per server parameter.

Engine Configuration Limit

- [Metrics](#)
- [Troubleshooting](#)

When a Sybase server begins to experience I/O delays, the result can be a very dissatisfied user community. When such problems begin to occur, things to investigate include the various Sybase or operating system limits. It could be that I/O operations are being blocked by one or both.

The Engine Configuration Limit statistic shows nonzero numbers if Sybase has exceeded its limit for the number of asynchronous disk I/O requests that can be outstanding for a Sybase engine at one time.

Metrics

Consistent numbers above zero should be a cause for concern.

Troubleshooting

If you continue to see nonzero numbers for this statistic, you can raise this limit using `sp_configure` with the `max async I/Os per engine` parameter.

Operating System Limit

- [Metrics](#)
- [Troubleshooting](#)

The user community can become very dissatisfied when a Sybase server begins to experience I/O delays. When such problems begin to occur, things to investigate include the Sybase or operating system limits. It could be that I/O operations are being blocked by one or both.

The Operating System Limit statistic shows nonzero numbers if Sybase has detected that the limit for asynchronous I/Os has been exceeded.

Metrics

You should be concerned if you consistently see numbers greater than zero.

Troubleshooting

In most UNIX operating systems, there is a kernel parameter that limits the number of asynchronous I/Os that can occur at one time. If you continue to see nonzero numbers for this statistic, you should look into raising this limit.

Deadlock Rate

- [Metrics](#)
- [Troubleshooting](#)

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process' page or object. Each waits for the other to release the necessary lock. Sybase constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates the transaction. The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

The deadlock rate displays the percentage of times deadlocks occurred vs. the percentage of locks requested and immediately granted.

Metrics

You should immediately investigate a percentage much above zero to prevent the situation from mushrooming. You can easily drill down and discover the exact process(es) holding locks and deadlocks that are blocking out other user activity.

Troubleshooting

Well-designed applications can minimize deadlocks by always acquiring locks in the same order. You should always do updates to multiple tables in the same order.

Once Sybase discovers a deadlock, it takes action and remedies the situation. Performance Center makes it easier to discover how prevalent deadlock activity is on a system; preventing deadlocks from occurring in the first place is more difficult.

Those responsible for writing systems can minimize deadlocks by ensuring that applications acquire locks in the same order. Likewise, you should always do updates and other DML that act on multiple tables in the same order.

You can also shrink the amount of time that Sybase waits to check for deadlocks by modifying the deadlock checking period configuration parameter.

Deadlocks

- [Metrics](#)
- [Troubleshooting](#)

A deadlock occurs when two processes have a lock on a separate page or object and each wants to acquire a lock on the other process' page or object. Each waits for the other to release the necessary lock. Sybase constantly checks for deadlocks and, when found, chooses the transaction that has accumulated the least amount of CPU time and terminates it (the transaction). The server then rolls back that transaction and issues a notification of the event. The other process gets to move forward.

The deadlock statistic displays the number of current deadlocks in a Sybase server.

Metrics

A nonzero number should clue you into the fact that application conflicts are likely being experienced by your user community.

Troubleshooting

Well-designed applications can minimize deadlocks by always acquiring locks in the same order. Updates to multiple tables should always be performed in the same order.

Once Sybase discovers a deadlock, it takes action and remedies the situation. Discovering how prevalent deadlock activity is on a system is made easier by using tools like Performance Center, but preventing deadlocks from occurring in the first place is more difficult.

Those responsible for writing systems can minimize deadlocks by ensuring that applications acquire locks in the same order. Likewise, updates and other DML that act on multiple tables should always be performed in the same order.

You can also shrink the amount of time that Sybase waits before performing a deadlock check by reducing the deadlock checking period parameter.

Current Locks

- [Metrics](#)

- [Troubleshooting](#)

There are varieties of operations in Sybase that require the use of locks. The Current Locks statistic represents the number of total locks currently active in Sybase.

TIP: Click this statistic to drill down to the [All Locks tab](#) of the Locks view.

NOTE: This statistic is also available on the [Users performance category view](#).

Metrics

The only thing to watch with respect to locks is if the number approaches the Sybase limit for available locks.

Troubleshooting

If the number of current locks in a Sybase server approaches the Sybase limit for available locks, you can look into increasing the Number of Locks configuration parameter.

Lock Contention

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although Sybase supports flexible locking mechanisms, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

The blocking lock rate displays the percentage of times blocks occurred vs. the percentage of locks requested and immediately granted.

Metrics

To prevent a mushrooming situation, you should immediately investigate any percentages greater than zero for this statistic.

You can easily drill down and discover the exact process(es) holding locks that are blocking out other user activity.

Troubleshooting

Once you discover a blocking lock situation, you can normally remedy it by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing, and usually results in other user processes completing in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in Sybase. You can change this behavior by modifying the lock wait period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

Blocked

- [Metrics](#)

- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Sybase supports flexible locking mechanisms, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

The Blocked statistic displays the number of current processes blocked by other processes.

TIP: Click this statistic to drill down to the [Blocking Locks tab](#) of the Locks view.

NOTE: This statistic is also available on the [Users performance category view](#).

Metrics

While the Lock Contention rate is a better measure of the overall lock contention situation, consistently seeing positive numbers for the Blocked statistic should also clue you into the fact that a bottleneck exists for some processes. You can easily drill down and discover the exact process(es) holding locks that are blocking other user activity.

Troubleshooting

Once you discover a blocking lock situation, you can normally remedy it by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing, and usually results in other user processes completing in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in Sybase. You can change this behavior by modifying the lock wait period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

Network Contention Rate

- [Metrics](#)
- [Troubleshooting](#)

The Sybase server normally sends and receives network packets at a regular rate. If the network begins to become saturated, Sybase could experience delays in network I/O. The Network Contention rate indicates the percentage of times network I/O activity was delayed. The following network contention rate statistics are available on the Contention performance category view:

- [Network Delays](#)
- [Network Requests](#)

Metrics

Seeing a network contention rate greater than 1% could indicate a challenged network structure.

Troubleshooting

Outside of ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase server.

Network Delays

- [Metrics](#)
- [Troubleshooting](#)

The Sybase server normally sends and receives network packets at a regular rate. If the network begins to become saturated, Sybase could experience delays in network I/O. The Network Delays statistic reflects the number of times network I/O activity was delayed.

NOTE: This statistic displays on both the [Network performance category view](#) and the Contention performance category view.

Metrics

Seeing a network contention rate greater than 1% could indicate a challenged network structure.

Troubleshooting

Beyond ensuring that the existing network is sufficient for handling the current load (database and nondatabase), other items to look at from a Sybase-specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase server.

Network Requests

Network Requests represents the total TDS packets received and sent by Sybase.

NOTE: This statistic displays on both the [Network performance category view](#) and the Contention performance category view.

Contention Detail View

The following tabbed pages are available on the Contention Detail view:

- [All Locks](#)
- [Blocking Locks](#)

All Locks Tab

- [Metrics](#)
- [Troubleshooting](#)

Data integrity is maintained in a database through the use of locks. There are many variations of locks that can be applied to data objects, with some being very restrictive. The All Locks tab of the Locks view presents granular information relating to locks that currently exist on the Sybase server. The table below describes the information available on the All Locks tab of the Locks view:

Column	Description
PID	The process of the view.
User Name	The login name of the user.
Database	The database that contains the locks.
Lock Type	The type of lock being applied.
Object Name	The object name being locked.
Status	The status of the transaction.
Lock Page	The page number where the lock is currently applied.
Lock Class	The name of the cursor the lock is associated with (if any).
Host	The name of the host computer.
Program	The program the process is running.
Command	The command being issued by the process.
CPU Time	The amount of CPU time accumulated for the current command.
Physical I/O	The amount of physical I/O accumulated for the current command.
Mem Usage	The amount of memory accumulated for the current command.
FID	The process ID of the worker process' parent.
Transaction	The name of the associated transaction (if any).

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

The main thing to watch for with respect to locks is that all DML locks currently held on the system do not approach the number of locks limit specified in the Sybase server's configuration. The parameter number of locks limits how many locks can exist on the system at one time.

Troubleshooting

If the total number of locks approaches the number of locks limit, you should:

- Ensure that user processes are efficiently using locks and are committing frequently to avoid excessive lock hold times before editing the current Sybase configuration.
- Edit the configuration file for the Sybase server.
- Increase the amount of number of locks to a higher value.
- Cycle the Sybase server when possible to allow the new value to take effect.

Blocking Locks Tab

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although Sybase supports flexible locking mechanisms, blocking lock situations crop up. User processes that hold exclude locks without releasing them via a proper COMMIT frequency most often cause blocks.

The Blocking Locks tab of the Locks view shows granular data concerning current blocking lock scenarios. The table below describes the information available on the Blocking Locks tab of the Locks view:

Column	Description
Holding PID	The process ID of the user holding the blocking lock.
Holding User	The login name of the user holding the blocking lock.
Waiting PID	The process ID of the user waiting for a lock.
Waiting User	The login name of the user waiting for a lock.
Database	The database that contains the locks.
Status	The status of the transaction.
Lock Type	The type of lock being applied.
Time Blocked	The amount of time that a process has been waiting for the lock, in seconds.
Lock Page	The page number where the lock is currently applied.
Lock Class	The name of the cursor the lock is associated with (if any).
Object Name	The object name being locked.
Holding Host	The name of the host computer with the blocking lock.
Waiting Host	The name of the host computer waiting for the lock.
Holding Program	The program the process is running that has the lock.
Waiting Program	The program the process is running that is waiting for the lock.
Holding Command	The command being issued by the process holding the lock.
Waiting Command	The command being issued by the process waiting for the lock.
CPU Time	The amount of CPU time accumulated for the current command.
Physical I/O	The amount of physical I/O accumulated for the current command.
Mem Usage	The amount of memory accumulated for the current command.
Holding FID	The process ID of the worker process' parent that has the lock.
Waiting FID	The process ID of the worker process' parent that is waiting for the lock.
Transaction	The name of the associated transaction (if any).

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

To prevent a mushrooming situation, you should investigate blocking locks that persist for noticeable periods of time, especially if the object(s) in question are hot objects for a database.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects he or she was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in Sybase. You can change this behavior by modifying the lock wait period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

Users Statistics - Sybase

The Users performance category view displays the following vital Sybase user statistics:

- [Active Connections](#)
- [Current Locks](#)
- [Inactive Connections](#)
- [Rows Deleted](#)
- [Rows Inserted](#)
- [Rows Updated](#)
- [Row Updated \(Data Locks Only\)](#)
- [Session Leaders - CPU](#)
- [Session Leaders - I/O](#)
- [Session Leaders - Memory](#)
- [Total Rows Affected](#)
- [Committed Transactions](#)
- [ULC Flushes](#)
- [Users Blocked](#)
- [Lock Promotions](#)

Lock Promotions

- [Metrics](#)
- [Troubleshooting](#)

The Lock Promotions value represents the average number of lock promotion types combined per second and per transaction.

Metrics

Lock promotions occur when the following escalations take place:

- "Ex-Page to Ex-Table" - Exclusive page to exclusive table.
- "Sh-Page to Sh-Table" - Shared page to shared table.
- "Ex-Row to Ex-Table" - Exclusive row to exclusive table.

- "Sh-R to Sh-Table - Shared row to shared table.
- "Sh-Next-Key to Sh-Table" - Shared next-key to shared table.

Troubleshooting

If lock contention is high and lock promotion is frequent, you should consider changing the lock promotion thresholds for the tables involved. You can configure the lock promotion threshold either server-wide or for individual tables.

Active Connections

The Active Connections statistic represents the total number of active and open threads reported in the Sybase server. This number displays the number of processes actively performing work.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

None.

Inactive Connections

- [Metrics](#)

The Inactive Connections statistic represents the total number of Sybase threads that are not actively running on one of the server engines.

TIP: Click this statistic to drill down to the [Processes tab](#) of the Users Detail view.

Metrics

Seeing a large number of inactive connections is not generally a cause for concern. If the threads are in a sleeping state (waiting for a resource or performing large volumes of disk I/O), further investigation can be warranted.

Current Locks

- [Metrics](#)
- [Troubleshooting](#)

The Current Lock statistic displays the total number of locks obtained/requested by processes in the database.

TIP: Click this statistic to drill down to the [All Locks tab](#) of the Locks view.

NOTE: This statistic is also available on the [Contention performance category view](#).

Metrics

The main thing to watch with respect to current locks is that all DML locks currently held on the system do not approach the Number of Locks limit specified in the Sybase server's configuration. The parameter for number of locks limits how many locks can exist on the system at once.

Troubleshooting

If the total number of locks approaches the number of locks limit, you should:

- 1 Ensure that user processes are efficiently using locks and are committing frequently to avoid excessive lock hold times before editing the current Sybase configuration.
- 2 Edit the configuration file for the Sybase server.
- 3 Increase the number of locks to a higher value.
- 4 Cycle the Sybase server when possible to allow the new value to take effect.

Users Blocked

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although Sybase supports flexible locking mechanisms, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

The blocked statistic displays the number of current processes blocked by other processes.

NOTE: Click this statistic to drill down to the [Blocking Locks](#) tab of the Locks view.

NOTE: This statistic is also available on the [Contention performance category view](#).

Metrics

While the lock contention rate is a better measure of the overall lock contention situation, consistently seeing positive numbers for the blocked statistic should clue you into the fact that there is a bottleneck for some processes. You can easily drill down and discover the exact process(es) holding locks that are blocking out other user activity.

Troubleshooting

Once you discover a blocking lock situation, you can normally remedy it by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing, and usually results in other user processes completing in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in Sybase. You can change this behavior by modifying the lock wait period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

Committed Transactions

- [Metrics](#)

Committed Transactions gives the number of transactions committed since the last refresh inside Performance Center. This includes transactions that meet explicit, implicit, and ANSI definitions for committed transactions. Note that multidatabase transactions are counted.

TIP: Click this statistic to drill down to the [Transactions tab](#) of the Users Detail view.

Metrics

Multidatabase transactions generally incur more overhead than single database transactions (for example, a transaction that modifies two databases is counted as two transactions). They usually involve more log activity and two-phase commits between the different databases, as well as cause more ULC flushes. You can improve performance by reducing the number of multidatabase transactions.

ULC Flushes

- [Metrics](#)

The total number of times that user log caches (ULCs) were flushed to a transaction log during the sample interval.

Metrics

None.

Rows Inserted

- [Metrics](#)
- [Troubleshooting](#)

The Rows Inserted statistic reflects the number of row inserts to heap and clustered tables. Note that it does not include the number of fast bulk copy inserts, because these are written directly to the data pages themselves.

TIP: Click this statistic to drill down to the [DML Detail tab](#) of the Users Detail view.

Metrics

Large numbers of inserts are no cause for concern. The only thing to keep an eye on is that heavy insert activity on heap tables can potentially cause lock contention. It is easy to investigate lock contention in the Performance Center's Locks view.

Troubleshooting

If insert activity seems to be causing lock contention, you can consider creating a clustered index that randomizes insert activity on the tables in question. Alternatives are establishing partitions on an unpartitioned table or increasing the number of partitions on a partitioned table.

Keep in mind that an unwanted by-product of clustered indexes is occasional page splitting.

Rows Updated

- [Metrics](#)

The Rows Updated statistic represents all deferred and direct update activity.

TIP: Click this statistic to drill down to the [DML Detail tab](#) of the Users Detail view.

Metrics

None.

Rows Updated (Data Only Locks)

- [Metrics](#)

The Rows Updated (Data Only Locks) statistic represents all deferred and direct update activity.

TIP: Click this statistic to drill down to the [DML Detail tab](#) of the Users Detail view.

Metrics

None.

Rows Deleted

- [Metrics](#)

The Rows Deleted statistic represents all deferred and direct delete activity.

TIP: Click this statistic to drill down to the [DML Detail tab](#) of the Users Detail view.

Metrics

None.

Total Rows Affected

- [Metrics](#)

The Total Rows Affected statistic represents all rows impacted by some form of DML activity (inserts, updates, deletes).

TIP: Click this statistic to drill down to the [DML Detail tab](#) of the Users Detail view.

Metrics

None.

Session Leaders - Memory

- [Metrics](#)
- [Troubleshooting](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problems could be runaway processes, untuned batch procedures, or user-initiated operations. Often, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system levels.

NOTE: This statistic displays on both the [Memory performance category view](#) and the Users performance category view.

TIP: Click this statistic to drill down to the [Leading Sessions tab](#) of the Memory Detail view.

Metrics

If your Sybase server does not have an overabundance of memory, you should check periodically to see who your heavy memory users are. You should also check the total percentage of memory each takes up. If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing.

Troubleshooting

You can use the Session Leaders - Memory statistic to find the users with the greatest current allocations of overall memory. Runaway processes can be immediately terminated from within the Performance Center Client.

Session Leaders - I/O

- [Metrics](#)

When a system undergoes heavy I/O activity, sometimes you find that all user connections are contributing equally to the overall load. More often than not, however, one or two user connections are responsible for 75% or more of the I/O activity. It can be that a large batch load or other typical process is running that is perfectly okay for your system. Alternatively, it can be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

The leading I/O session's display allows you to see who runs the leading sessions in your system with respect to I/O.

NOTE: This statistic displays on both the [I/O performance category view](#) and the Users performance category view.

TIP: Click this statistic to drill down to the [Leading Sessions tab](#) of the I/O Detail view.

Metrics

Finding one or two users who consume more than 75% of the total I/O load can indicate runaway or improper processes. By drilling down into the I/O activity of all users, you can quickly see if this is the case.

Session Leaders - CPU

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Often, user connections can get out of hand with CPU use, and extreme cases have caused headaches at both the database and operating system levels.

The leading CPU session's display allows you to see who runs the leading sessions are in your system with respect to CPU usage.

Metrics

Finding one or two users who use the majority of the CPU can indicate runaway or improper processes. By drilling down into the CPU activity of all users, you can quickly see if this is the case.

Users Detail View

The following tabbed pages are available on the Users Detail view:

- [Applications](#)
- [Blocking Locks](#)
- [DML Detail](#)
- [Locks](#)
- [Processes](#)
- [Transactions](#)

Top Memory Processes

- [Metrics](#)

The Top Memory Process statistic identifies the Sybase process that currently is using the highest percentage of memory in the database.

Metrics

None.

Applications Tab

Different programs/applications can access the Sybase server at the same time, with some being more resource-intensive than others. The Applications Detail tab on the Users Detail view summarizes resource consumption at the application level. The table below describes the information available on the Applications tab of the Users Detail view:

Column	Description
Application	The application name. NOTE: This can be blank if Sybase cannot determine the application accessing the server.
Application Count	The total number of like applications accessing the server.
Active	The count of active applications accessing the server.
Inactive	The count of inactive applications accessing the server.
Memory Used	The total amount of memory used by each application's current command.
CPU Used	The total amount of CPU used by each application's current command.
Physical I/O	The total amount of Physical I/O used by each application's current command.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

None.

Processes Tab

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Often, user connections can get out of hand with memory consumption, CPU, or physical I/O, and extreme cases have caused headaches at both the database and operating system levels.

The Processes tab of the Users Detail view displays information to help pinpoint problem processes. The table below describes the information available on the Processes tab of the Users Detail view:

Column	Description
User Name	The logon name the session is using.
PID	The process ID.
FID	The process ID of the worker process' parent.
Database	The database the process is attached to.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The client machine name the session is using.
Program	The executable the process is using against the Sybase server.
Memory	The amount of memory currently allocated to the process.
CPU	The amount of CPU currently in use by the process.
Physical I/O	The amount of physical I/O currently accumulated by the process.
Blocked	Whether the process is blocked by another process.
Command	The command the process is currently issuing.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

The key to spotting problem processes is to view their information in light of the total activity on a server. For example, if your database server does not have an overabundance of memory, you should periodically check to see who your heavy memory users are. You should also check the total percentage of memory each takes up. If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing and take action if necessary.

Locks Tab

- [Metrics](#)
- [Troubleshooting](#)

Data integrity is maintained in a database through the use of locks. There are many variations of locks that can be applied to data objects, with some being very restrictive. The Locks tab of the Users Detail view presents granular information relating to locks that currently exist on the Sybase server. The table below describes the information available on the Locks tab of the Users Detail view:

Column	Description
PID	The process of the user.
User Name	The login name of the user.
Database	The database that contains the locks.
Lock Type	The type of lock being applied.
Object Name	The object name being locked.
Status	The status of the transaction.
Lock Page	The page number where the lock is currently applied.
Lock Class	The name of the cursor the lock is associated with (if any).
Host	The name of the host computer.
Program	The program the process is running.
Command	The command being issued by the process.
CPU Time	The amount of CPU time accumulated for the current command.
Physical I/O	The amount of physical I/O accumulated for the current command.
Mem Usage	The amount of memory accumulated for the current command.
FID	The process ID of the worker process' parent.
Transaction	The name of the associated transaction (if any).

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

The main thing to watch with respect to current locks is that all DML locks currently held on the system do not approach the number of locks limit specified in the Sybase server's configuration. The parameter number of locks limits how many locks can exist on the system at once.

Troubleshooting

If the total number of locks approaches the number of locks limit, then:

- Before editing the current Sybase configuration, ensure that user processes are efficiently using locks and are committing frequently to avoid excessive lock hold times.
- Edit the configuration file for the Sybase server.
- Increase the amount of number of locks to a higher value.
- Cycle the Sybase server when possible to allow the new value to take effect.

Blocking Locks Tab

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Sybase supports flexible locking mechanisms, blocking lock situations do crop up. User processes holding exclusive locks and not releasing them via a proper COMMIT frequency most often cause blocks.

The Blocking Locks tab of the Users Detail view shows granular data concerning current blocking lock scenarios. The table below describes the information available on the Blocking Locks tab of the Users Detail view:

Column	Description
Holding PID	The process ID of the user holding the blocking lock.
Holding User	The login name of the user holding the blocking lock.
Waiting PID	The process ID of the user waiting for a lock.
Waiting User	The login name of the user waiting for a lock.
Database	The database that contains the locks.
Object Name	The object name being locked.
Status	The status of the transaction.
Lock Type	The type of lock being applied.
Time Blocked	The amount of time that a process has been waiting for the lock, in seconds.
Lock Page	The page number where the lock is currently applied.
Lock Class	The name of the cursor the lock is associated with (if any).
Holding Host	The name of the host computer with the blocking lock.
Waiting Host	The name of the host computer waiting for the lock.
Holding Program	The program the process is running that has the lock.
Waiting Program	The program the process is running that is waiting for the lock.
Holding Command	The command being issued by the process holding the lock.
Waiting Command	The command being issued by the process waiting for the lock.
CPU Time	The amount of CPU time accumulated for the current command.
Physical I/O	The amount of physical I/O accumulated for the current command.
Mem Usage	The amount of memory accumulated for the current command.
Holding FID	The process ID of the worker process' parent that has the lock.
Waiting FID	The process ID of the worker process' parent that is waiting for the lock.
Transaction	The name of the associated transaction (if any).

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

To prevent a mushrooming situation, you should investigate blocking locks that persist for noticeable periods of time should, especially if the object(s) in question are hot objects for a database.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects he or she was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is where it gets tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in Sybase. You can change this behavior by modifying the lock wait period configuration parameter, which limits the number of seconds that a process waits for a lock before timing out.

DML Detail Tab

- [Metrics](#)

Dynamic Sybase environments typically have large amounts of DML (data manipulation language) activity that occurs. Knowing the volume of INSERT, UPDATE, and DELETE activity can help you make correct decisions regarding sizing, object options, etc. The DML Detail tab of the Users Detail view provides a quick snapshot of the amount of object/data manipulation that is occurring. The table below describes details of the DML Detail tab of the Users Detail view:

Detail	Description
Activity Type	The type of INSERT, UPDATE, or DELETE.
Amount	The number of operations that have taken place since the last refresh.

The table below describes the statistics available on the DML Detail tab of the Users Detail view:

Statistic	Description
Heap Table Inserts	Total rows inserted into heap tables during sample interval.
Clustered Table Inserts	Total rows inserted into clustered tables during sample interval.
Deferred Updates	Total number of updates that were accomplished in two steps during sample interval.
Direct In Place Updates	Total number of in-place updates (no data rows are moved on the data page) during sample interval.
Direct Cheap Updates	Total number of on-page updates (the length of the data row changes) during sample interval.
Direct Expensive Updates	Total number of delete/insert direct updates (rows are deleted from their original location and inserted at a new location) during sample interval.
Deferred Deletes	Total rows deleted during the sample interval.
Direct Deletes	Total rows directly deleted during the sample interval.

Metrics

Response times can differ greatly for various types of DML activity. For example, deferred UPDATE and DELETE operations (which are often caused by join conditions or because referential integrity conditions exist) normally perform slower than direct UPDATE and DELETEs. Deferred operations can involve rereading the transaction log to finalize the operation whereas direct operations have no such need. Another example would be that a direct expensive update involves additional index locks (and is therefore more restrictive), while direct cheap updates do not.

Transactions Tab

- [Metrics](#)

Transactions are used in Sybase to preserve data integrity and keep logical units of work together. Long-running transactions, however, run the risk of holding data and object locks longer than necessary and possibly degrading overall system response times. The Transactions tab of the Users Detail view shows all running transactions in a database and their current state. The table below describes the information available on the Transactions tab of the Users Detail view:

Column	Description
SPID	The process ID.
Login	The login name of the user initiating the transaction.
Database	The database involving the transaction.
Status	The status of the transaction.
Start Time	The time the transaction began.
Type	The type of transaction (local, remote, etc.)
State	The point where the transaction is (In Command, Committed, etc.)
Command	The current command being issued.
Connection	Whether the transaction is current attached or detached.
CPU	The amount of CPU currently being used by the transaction.
Memory	The amount of memory used by the current transaction.
Physical I/O	The current cumulative number of reads and writes issued by the process.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

You should examine transactions that have been running for excessive periods of time to see if they are being blocked by other transactions. If you find this is the case, Performance Center offers mechanisms that allow you to kill any process that is blocking another. If blocking lock activity is occurring, a small redesign of the application or more frequent COMMIT TRANSACTION points can prove to be useful preventive measures.

Network Statistics - Sybase

The Network performance category view displays the following vital network statistics:

- [Total Packets Received](#)
- [Total Packets Sent](#)
- [Total Bytes Received](#)
- [Total Bytes Sent](#)
- [Network Errors](#)
- [Network Contention Rate](#)
- [Network Delays](#)
- [Network Requests](#)

Average Bytes Received Per Packet/Average Bytes Sent Per Packet

These statistics represent the average number of bytes received or sent per packet, respectively, by the Sybase Adaptive Server engine during the sample interval.

Metrics

None.

Total Packets Received

- [Metrics](#)

Total Packets Received reflects the number of times Sybase received a packet from a client application.

Metrics

None.

Total Packets Sent

- [Metrics](#)

Total Packets Sent reflects the number of times Sybase sent a packet to a client application.

Metrics

None.

Total Bytes Received

- [Metrics](#)

Total Bytes Received reflects the number of bytes received by Sybase since the last refresh in Performance Center.

Metrics

None.

Total Bytes Sent

- [Metrics](#)

Total Bytes Sent reflects the number of bytes sent to Sybase since the last refresh in Performance Center.

Metrics

None.

Network Errors

- [Metrics](#)

- [Troubleshooting](#)

The Network Errors statistic reflects the number of times that network errors were detected by Sybase while reading and writing packets.

Metrics

Seeing a consistent value much above zero could indicate a challenged network structure.

Troubleshooting

Beyond ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase-specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase server.

Network Contention Rate

- [Metrics](#)
- [Troubleshooting](#)

The Sybase server normally sends and receives network packets at a regular rate. If the network begins to become saturated, Sybase can experience delays in network I/O. The network contention rate indicates the percentage of times network I/O activity was delayed.

Metrics

Seeing a network contention rate greater than 1% could indicate a challenged network structure.

Troubleshooting

Outside of ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase-specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase server.

Network Delays

- [Metrics](#)
- [Troubleshooting](#)

The Sybase server normally sends and receives network packets at a regular rate. If the network begins to become saturated, Sybase can experience delays in network I/O. The Network Delays statistic reflects the number of times network I/O activity was delayed.

Metrics

Seeing a network contention rate greater than 1% could indicate a challenged network structure.

Troubleshooting

Beyond ensuring that the existing network is sufficient for handling the current load (database and non-database), other items to look at from a Sybase-specific standpoint include validating that only the necessary amount of data is being sent and returned to requesting users. You can also examine the default packet size and see if it is too small for the average packet size being sent/received by the Sybase server.

Network Requests

Network Requests represents the total TDS packets received and sent by Sybase.

Metrics

None.

OS Page Statistics

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Center lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX
- Linux
- Solaris
- Unix
- Windows XP, 2000, and NT

Summary Tab

The OS Summary tab displays the following statistics to communicate the general overall performance levels of the operating system:

- [Disk Time](#)
- Load Average
- [Processor Time](#)
- [Paged Memory Used \(Windows\)](#)
- [Swap Memory Used \(AIX, HP-UX, Linux, Solaris, Unix\)](#)
- [Average Disk Queue](#)
- [Network Output Queue \(Windows\)](#)
- [Network Queue \(Solaris\)](#)
- [Page Faults/Sec](#)
- [Processor Queue](#)
- Processor Speed
- Processor
- [Available Paged Memory \(Windows\)](#)
- [Available Physical Memory](#)
- [Available Swap Memory \(AIX, HP-UX, Linux, Solaris, Unix\)](#)

- [Total Paged Memory \(Windows\)](#)
- [Total Physical Memory](#)
- [Total Swap Memory \(AIX, HP-UX, Linux, Solaris, Unix\)](#)
- [Free Disk Space](#)
- Total Disk Space
- [Used Disk Space](#)
- [Number of Logins](#)
- [Number of Processes](#)
- Number of Processors
- [Top CPU Process](#)
- [Top I/O Process](#)
- [Top Memory Process](#)

Processor Time

The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

Metrics

If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.

Troubleshooting

Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

Processor Speed

The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Processor

The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Disk Time

The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

Metrics

You should avoid consistently seeing values for this statistic greater than 90%.

Troubleshooting

Add more disk drives and partition the files among all of the drives.

Load Average

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Metrics

High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Paged Memory Used

The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

NOTE: This statistic is available for the Windows platform.

Metrics

This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.

Troubleshooting

Increase the size of page file.

Number of Processors

This statistic displays the number of processors currently in use.

Swap Memory Used

The Swap Memory Used statistic is the percentage of swap space currently in use.

Metrics

If the percentage of swap memory used is above 90%, you may be running out of memory.

Troubleshooting

Increase the size of your swap files.

Average Disk Queue

The Average Disk Queue statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Metrics

This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.

Troubleshooting

Some things you can do if you have problems with this statistic include:

- Redistribute the data on the disk with the large average disk queue to other disks.
- Upgrade to faster disk(s).

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics

This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue

The Processor Queue Length statistic is the number of threads in the processor queue.

Metrics

Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting

A sustained high value in the Processor Queue could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Network Output Queue/Network Queue

The Network Output Queue Length statistic is the number of threads in the processor queue.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics

Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting

A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Available Physical Memory

The Available Physical Memory statistic represents the amount of RAM available to all processes.

Metrics

This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Troubleshooting

Some things you can do if you have problems with this statistic include:

- Check the running processes to see if there are any memory leaks.
- Stop any services that are not required.
- Install additional RAM.

Available Paged Memory

The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

NOTE: This statistic is available for the Windows platform.

Metrics

If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.

Troubleshooting

Increase the size of page file.

Available Swap Memory

The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Metrics

If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.

Troubleshooting

Increase the size of swap files.

Total Physical Memory

The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Metrics

This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Total Paged Memory/Total Swap Memory

The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics

It is recommended that this value is between 1.5 to 3 times the amount of RAM on the system.

Used Disk Space

The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Total Disk Space

Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two:

- 1 If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- 2 You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Free Disk Space

The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Top Memory Process

Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics

If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Processes Overview

The Processes Overview of the OS Summary includes the following sections:

- [Top CPU Process](#)
- [Top I/O Process](#)
- [Top Memory Process](#)

Top CPU Process

Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics

If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.

Troubleshooting

Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process

The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Number of Logins

This statistic displays the total number of logins on the server.

Number of Processes

This statistic displays the total number of processes on the server.

CPU Tab

The CPU tab of the OS Detail includes the following sections:

- [Context Switches/Sec](#)
- [CPU Utilization](#)
- [Interrupts/Sec](#)
- [Processor Queue Length](#)

CPU Utilization

The CPU Utilization section includes the following information:

- [% Privileged Time](#)
- [% User Time](#)

% Privileged Time

The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

NOTE: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

Metrics

The ideal range should be 0-40% (less than 40% indicates excessive system activity).

Troubleshooting

If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

% User Time

The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Metrics

If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

Interrupts/Sec

The Interrupts/Sec statistic is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. It does not include deferred procedure calls (DPCs), which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every ten milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Metrics

The ideal range should be 0-5000. A number greater than 5000 indicates possible excessive hardware interrupts; justification is dependent on device activity.

Context Switches/Sec

The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Metrics

The ideal range should be between 0-10,000. A number greater than 10,000 may indicate too many threads contending for resources.

Processor Queue Length

The Processor Queue Length statistic is the number of threads in the processor queue. There is a single queue for processor time even on computers with multiple processors.

NOTE: For Windows systems, unlike the disk counters, this counter shows ready threads only, not threads that are running.

Metrics

A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Processes Tab

The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

Column	Description
Process	The name of the process.
User	The user of the process.
ID	The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
CPU	The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.
User Mode	The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.
Memory WINDOWS ONLY	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory (MB)	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory	Memory is the percentage of the memory used of the total memory.
Active Memory	Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.
I/O Data	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
Elapsed Time	The total elapsed time, in seconds, that this process has been running.
Thread Count	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Handle Count	The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.
Priority	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Creating Proc ID	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Page Faults/Sec	Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.
Page File	Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private	Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.

I/O Tab

The table below describes the information available in this section:

Column	Description
Disk	The disk number assignment.
Reading (KB/s)	The amount of bytes read from the device.
Writing (KB/s)	The amount of bytes written to the device.
Disk Read Time	Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Disk Write Time	Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Disk Time	Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.
Avg. Read Queue	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
Avg. Write Queue	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Disk Reads/Sec	Disk Reads/Sec is the rate of read operations on the disk.
Disk Writes/Sec	Disk Writes/Sec is the rate of write operations on the disk.

Memory Tab

The Memory tab of the OS Detail page includes the following sections:

- [Cache Efficiency](#)
- [Free Physical](#)
- [Free Paged](#)
- [Paging Activity](#)
- [Page Faults](#)
- [Total Physical](#)
- [Total Paged](#)

Paging Activity

The Paging Activity section includes the following statistics:

- [Pages Input/Sec](#)
- [Pages Output/Sec](#)

Pages Input/Sec

The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Metrics

This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Pages Output/Sec

The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Metrics

Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Free Physical

The Free Physical statistic is the amount of physical memory that is uncommitted.

Metrics

None.

Free Paged

The Free Paged statistic is the amount of uncommitted virtual memory.

Metrics

None.

Total Physical

The Total Physical statistic is the total physical memory available.

Metrics

None.

Total Paged

The Total Paged statistic is the amount of total virtual memory, in bytes. Used Memory is the physical memory that has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.

Metrics

None.

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics

This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Cache Efficiency

The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

- [Copy Read Hits%](#)
- [Data Map Hits%](#)
- [MDL Read Hits%](#)
- [Pin Read Hits%](#)

Copy Read Hits %

The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

Metrics

When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Data Map Hits %

The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %

The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %

The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab

The Space tab of the OS Detail page includes the following sections:

- [Disk Space Free](#)
- [Disk Space Detail](#)

Disk Space Free

The Disk Space Free metric displays the amount of free disk space in megabytes.

Metric

None.

Disk Space Detail

The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.

Network Tab

The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

Column	Description
Network Interface	The name of network interface.
INET Address/Address	The IP address assigned to the network interface.
Pkts Sent/Sec	The number of packets sent per second.
Pkts Received/Sec	The number of packets received per second.
Sent (KB/Sec)	The number of bytes sent per second.
Received (KB/Sec)	The number of bytes received per second.
Out Pkts Discarded	The number of outbound packets discarded.
In Pkts Discarded	The number of inbound packets discarded.
Out Pkt Errors	The number of outbound packet errors.
In Pkt Errors	The number of inbound packet errors.
Queue Length	The queue length.
Collisions	The number of collisions.
Packets Discarded	The number of packets discarded.

Oracle Expert Guide

This section includes expert help for all Oracle categories and statistics in Performance Center views. For detailed information on using the application, see [Using Performance Center](#). This guide includes the following sections:

- [Home View Statistics](#)
- [Contention Page Statistics](#)
- [I/O Page Statistics](#)
- [Memory Page Statistics](#)
- [Network Statistics](#)
- [Objects Page Statistics](#)
- [OS Page Statistics](#)
- [Session Details](#)
- [Space Page Statistics](#)
- [Users Page Statistics](#)
- [Other Statistics](#)

Home Statistics

The Home view lets you review availability and overall performance of all monitored databases from a single window. The Home page includes the following sections:

- [Contention Vital Signs](#)
- [I/O Vital Signs](#)
- [Memory Vital Signs](#)
- [Network Vital Signs](#)
- [Space Vital Signs](#)
- [Users Vital Signs](#)

Related Topics

[I/O Page Statistics](#)

[Memory Page Statistics](#)

[Objects Page Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

[Users Page Statistics](#)

[Network Statistics](#)

[Contention Statistics](#)

[Other Statistics](#)

Memory Vital Signs

The following memory statistics are on the Oracle Home view:

- [Buffer Cache](#)
- [Dictionary Cache](#)
- [Library Cache](#)
- [Memory Sort Ratio](#)

Contention Vital Signs

The following contention statistics are on the Oracle Home view:

- [Enqueue Waits](#)
- [Redo Log Space Wait Time](#)
- [Users Blocked](#)
- [Users Waiting](#)

Users Waiting

- [Metrics](#)
- [Troubleshooting](#)

User connections that are waiting on a system generally occur for two reasons:

- 1 A process waits because a requested resource is not available.
- 2 A process waits for Oracle to perform a prerequisite task for its given operation.

These two wait causes are worth your time and investigation; idle waits (processes waiting because they have no work) should not worry you at all.

Metrics

To determine the actual wait causes user connections are experiencing, drill down from the global count of users waiting into the actual system and user wait details of your target database. This lets you locate the exact causes of currently experienced waits.

Troubleshooting

If you find a problem, drill down into wait details to determine whether the waits are resource related.

Users Blocked

- [Metrics](#)
- [Troubleshooting](#)

On a small system, a single blocking user has the potential to stop work for nearly all other processes. On larger systems, this can cause major headaches. Although Oracle supports unlimited row-level locking, blocking lock situations do occur. User processes holding exclusive locks and not releasing them via a proper COMMIT frequency, generally cause most blocks.

Metrics

You should investigate any blocking lock statistic indicator above zero to prevent a mushrooming situation.

Troubleshooting

You can quickly remedy a blocking lock situation by issuing a KILL against the offending process. This eliminates the user's stranglehold on the accessed objects and lets other user processes complete. Tools like Performance Center make it easier to discover the blocked lock situation, the tricky part is preventing the blocking lock situation in the first place.

The culprit of blocking lock scenarios is usually the application design, or the SQL used within the application itself. Properly coding an application to reference database objects in an efficient order and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT...FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible, something not always easy to do.

Enqueue Waits

- [Metrics](#)
- [Troubleshooting](#)

An enqueue is an advanced locking device that lets multiple database processes share certain resources. Enqueue waits typically occur when sessions wait to be granted a requested lock. Sometimes these locks are internal Oracle locks while other times they could be locks for rows of data in a table.

NOTE: Enqueues are issued implicitly by Oracle.

Metrics

You should investigate any enqueue waits that read consistently above one or more (delta statistics).

Troubleshooting

Removing contention for enqueues is usually an application design issue. Many enqueue waits are either contention for certain rows in the database or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that your SQL does not tarry over rows in the database during modification operations. If it does, you should tune your SQL.

Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).

Redo Log Space Wait Time

- [Metrics](#)

- [Troubleshooting](#)

The Oracle RDBMS is able to manage recovery by recording all changes made to a database through the use of redo log files. Oracle writes modifications made to a database to the redo log files, which you can archive off to another medium for disaster recovery. The background process that performs these operations is Oracle's Log Writer (LGWR). There is a buffer area in Oracle's System Global Area (SGA) that is used to reduce redo log file I/O, whose size, or lack thereof, can affect performance in a busy system. Sometimes a user process must wait for space in the redo log buffer. Oracle uses the log buffer to cache redo entries prior to writing them to disk. If the buffer area is not large enough for the redo entry load, waits can occur. While waits can be an important indicator of contention, a better measure is the amount of wait time that your users experience.

Metrics

The two main numbers to watch are:

- 1 Redo log space requests
- 2 Redo log wait time

If either statistic strays too far from zero, you may want to increase the `log_buffer` parameter and add more memory to the redo log buffer.

Troubleshooting

If you find a problem, do the following:

- 1 Edit the `Init.ora` file for the database.
- 2 Increase the amount of `log_buffer` to a higher value (take care not to over-allocate; ensure enough free memory exists on server before increasing value).
- 3 Cycle the Oracle server when possible to allow the new value to take effect.
- 4 Monitor new value to see if performance improves.

When adjusting the `log_buffer`, make sure that the amount is a multiple of the block size. Otherwise, on database startup, Oracle returns an error stating that you have entered an invalid amount for the redo log buffer.

NOTE: If you make the `log_buffer` parameter smaller than its default size for a given platform, Oracle will silently increase it.

I/O Vital Signs

The following I/O statistics are on the Oracle Home view:

- [Logical Changes](#)
- [Logical Reads](#)
- [Physical Reads](#)
- [Physical Writes](#)

Physical Reads

This value reflects total number of physical reads performed on all datafiles since the last refresh.

Metrics

Large numbers of physical reads may indicate that your buffer cache is too small. You should examine the buffer cache hit ratio to determine the overall effectiveness of logical vs. physical I/O.

Physical Writes

This value reflects total number of times the DBWR process has performed writes to various database datafiles since the last refresh.

Metrics

Wait events related to I/O activity are good indicators of physical I/O problems. These events include db file parallel write and db file single write. You can view both of these events in Performance Center.

Logical Reads

This value reflects total number of db block gets and consistent gets (data read from memory) since the last refresh.

Metrics

Large numbers of logical reads may not indicate a problem, although extremely large amounts could indicate a runaway process. You should examine the buffer cache hit ratio to determine the overall effectiveness of logical vs. physical I/O. You could also examine the User I/O Detail view to see if any one process is churning up a lot of logical I/O.

Logical Changes

This is the total number of changes that were made to all blocks in the SGA that were part of an update or delete operation. These changes generate redo log entries and are permanent if the transaction is committed. The number of logical changes is an approximate indication of total database work.

Metrics

None.

Users Vital Signs

The following users statistics are on the Oracle Home view:

- [Active Connections](#)
- [Current Locks](#)
- [Open Cursors](#)
- [Total Connections](#)

Total Connections

- [Metrics](#)
- [Troubleshooting](#)

This statistic represents the total number of open threads, or connections, currently reported in the database. This number includes both active and inactive processes.

Metrics

You should view the total number of sessions in light of the maximum number of processes allowed to connect to Oracle. The processes parameter in the Init.ora file specifies the maximum number of operating system user processes that can simultaneously connect to an Oracle Server. Checking the Users performance category view can show just how close the system is coming to running out of processes.

Troubleshooting

If the total number of connections approaches the processes limit, do the following:

- 1 Edit the Init.ora file for the database.
- 2 Increase the amount of processes to a higher value.
- 3 Cycle the Oracle server when possible to allow the new value to take effect.

Active Connections

This statistic represents the total number of active and open threads, or connections, reported in the database. The number displays the number of processes actively performing work.

Metrics

None.

Current Locks

- [Metrics](#)
- [Troubleshooting](#)

This statistic displays the total number of locks obtained by processes in the database.

Metrics

With respect to locks, the main thing to watch is that all DML locks currently held on the system do not approach the dml_locks limit specified in the Init.ora file. The parameter, dml_locks, limits how many locks can exist on the system at one time.

Troubleshooting

If the total number of locks approaches the dml_locks limit, do the following:

- 1 Ensure that user processes are efficiently using locks and are committing frequently to avoid excessive lock hold times before editing the Init.ora file.
- 2 Edit the Init.ora file for the database.
- 3 Increase the amount of dml_locks to a higher value.
- 4 Cycle the Oracle server when possible to allow the new value to take effect.

Space Vital Signs

The following space statistics are on the Oracle Home view:

- [Free Space](#)
- [Used Space](#)
- [Tablespace Low on Space](#)

Used Space and Free Space

- [Metrics](#)
- [Troubleshooting](#)

These statistics represent the total used and free space available in all tablespaces/datafiles in the database. Although this is good to know, you need a more detailed listing by tablespace to determine where any actual space shortages exist in the database.

You can view this information in the Performance Center Space performance category view.

Metrics

If any one tablespace begins to approach 90% used, the DBA should take action to prevent any future space allocation errors.

Troubleshooting

There are a couple of things a DBA can do to prevent a tablespace from running out of available free space:

- 1 Turn AUTOEXTEND on for the underlying tablespace's datafiles. This allows them to automatically grow when free space in the datafile has been exhausted.
- 2 Using the ALTER TABLESPACE...ADD DATAFILE... command, you can manually add a new datafile to a tablespace that is about to run out of available free space.

Tablespace Low on Space

- [Metrics](#)
- [Troubleshooting](#)

This statistic gives you a count of tablespaces with less than 10% free space.

Metrics

If any tablespace free space percent amount goes below 10%, you should take action to ensure that the tablespace does not run out of available free space.

NOTE: The default notification for Tablespace Low on Space is the designated Emergency Contact. If you do not want the notification to go to the designated Emergency Contact, you should create a Tablespace Low on Space notification.

Troubleshooting

There are two things you can do to ensure that a tablespace does not run out of available free space:

- 1 First, you should look into the use of Oracle's AUTOEXTEND feature. AUTOEXTEND lets you give an Oracle tablespace the ability to auto-grow when it has exhausted the free space contained within. You can let a tablespace grow in an unlimited fashion or put constraints on it to stop at a certain point. You can also dictate how much more free space the tablespace gets each time it needs more space than is available. However, AUTOEXTEND enabled for a tablespace does not mean that you cannot run out of space. Remember you still have the physical server limitations to contend with. Make sure you (or your sysadmin) keep a careful eye on the server drives that house your Oracle database files for available free space.
- 2 If the free space on a server drive nears its limit, disable AUTOEXTEND for the datafile(s) that are on that drive, and use the old-fashioned ALTER TABLESPACE... ADD DATAFILE command to place a new datafile for the tablespace on another drive that has more free space to offer.

TIP: AUTOEXTEND is not a replacement for proactive space planning on the part of the DBA. When the database needs extra space and AUTOEXTEND is activated by Oracle, you take a performance as Oracle allocates more space for the tablespace. Avoiding this type of extension aids performance, albeit in a small way.

Network Vital Signs

The following network statistics are on the Oracle Home view:

- [Bytes from Client](#)
- [Bytes from DBLink](#)
- [Bytes to Client](#)
- [Bytes to DBLink](#)

Bytes to Client

This statistic is the total number of bytes sent over the network to all Oracle client machines from the database since the last refresh.

Metrics

All users connected to a database via a client/server connection generate network traffic, so seeing decent volumes of network traffic is normal. Out of the ordinary numbers should raise alarms as possible network saturation could occur.

Bytes from Client

This statistic is the total number of bytes sent over the network to the Oracle database from all client machines since the last refresh.

Metrics

All users connected to a database via a client/server connection generate network traffic, so seeing decent volumes of network traffic is normal. Numbers out of the ordinary should raise alarms as possible network saturation could occur.

Bytes to DBLink

This statistic is the total number of bytes sent by all client machines over the network to any database link since the last refresh.

Metrics

All users connected to a database via a client/server connection generate network traffic, so seeing decent volumes of network traffic is normal. Numbers out of the ordinary should raise alarms as possible network saturation could occur.

NOTE: Database link traffic could experience longer response times depending on the distributed nature and setup of the involved databases.

Bytes from DBLink

This statistic is the total number of bytes received by all client machines over the network from any database link since the last refresh.

Metrics

All users connected to a database via a client/server connection generate network traffic, so seeing decent volumes of network traffic is normal. Numbers out of the ordinary should raise alarms as possible network saturation could occur.

NOTE: Database link traffic could experience longer response times depending on the distributed nature and setup of the involved databases.

- [Metrics](#)
- [Troubleshooting](#)

The buffer cache hit ratio is an indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. Data read from memory produces user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the Oracle buffer cache's purposes in life.

The table below describes the key counters Performance Analyst uses to calculate the buffer cache hit ratio:

Key Counter	Description
DB BLOCK GETS	Data read from memory for DML operations.
CONSISTENT GETS	Data read from rollback segments in memory.
PHYSICAL READS	Data read physically from disk.
Direct Reads	Data read physically from disk that bypasses the buffer cache. Direct reads are filtered out of overall physical reads so an accurate cache hit ratio can be determined.

Dividing the data read from memory by data read from disk yields the cache hit ratio.

NOTE: This statistic is also available on the [Memory home page](#).

Metrics

To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, every database has its own 'personality' and can exhibit excellent performance with below average readings for the cache hit ratio. Excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

Investigate consistent low readings of 60% or less.

NOTE: For Oracle8i or earlier, the adjustment of the `db_block_buffers` tuning parameter is required. For Oracle9i and later, the `db_cache_size` parameter is the parameter that needs attention. Any increases in `db_block_buffers` to take effect on Oracle8i or earlier, the database must be cycled. The `db_cache_size` parameter in Oracle9i or later, however, is dynamic and can be altered without stopping and starting the database instance.

Troubleshooting

If a problem is found in Oracle8i or earlier, do the following:

- Edit the `Init.ora` file for the database.
- Increase the amount of `db_block_buffers` to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later, do the following:

- Increase the size of the `db_cache_size` parameter through use of the `ALTER SYSTEM SET db_cache_size` command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value).
- Monitor the new value to see if performance improves.
- If using an `SPFILE`, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

Library Cache Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

Oracle's shared pool offers a number of tuning possibilities and is made up of two main memory areas:

- Library Cache
- Data Dictionary Cache

The library cache hit ratio offers a key indicator in determining the performance of the shared pool. It holds commonly used SQL statements - basically database code objects. The library cache hit ratio shows how often SQL code is reused by other database users vs. the number of times an SQL statement is broken down, parsed, and then loaded (or reloaded) into the shared pool.

You can improve performance by encouraging the reuse of SQL statements so expensive parse operations can be avoided. The library cache assists this tuning effort.

NOTE: This statistic is available on the Home page, the Memory home page and on the Library Cache tab of the Memory Detail.

Metrics

A high library cache hit ratio is a desirable thing. Strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse.

NOTE: When a database is first started, the library cache hit ratio is not at an optimal level because all code being used is relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, increase the `shared_pool_size` parameter.

Other red flags that can indicate a too small shared pool include:

- A wait count for the event 'latch free' of 10 or greater.
- The library cache wait count of two or greater.

These indicators can be tracked with Performance Analyst's Bottleneck and Wait detail views.

You can improve the library cache hit ratio by encouraging SQL code reuse through the implementation of bind variables. Discouraging hard coded literals in application code and instead making use of variables bound at run time aids in the reuse of SQL code that is maintained in Oracle's shared pool.

NOTE: Bind variables can have an affect on the cost-based optimizer though.

A second way is to pin frequently used code objects in memory so they are available when needed, by using the system supplied `DBMS_SHARED_POOL` package. You can use Performance Analyst to view objects in the shared pool that are always present and/or have increasing reload numbers to help identify objects that are good candidates for pinning.

Troubleshooting

If a problem is found in Oracle8i or earlier, do the following:

- Edit the `Init.ora` file for the database.
- Increase the amount of `shared_pool_size` to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later, do the following:

- Increase the size of the `shared_pool_size` parameter through use of the `ALTER SYSTEM SET shared_pool_size` command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an `SPFILE`, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

If you determine that SQL literals are causing SQL to not be reused, do the following (in Oracle 8.1.6 and later):

- Change the `cursor_sharing` parameter to `FORCE` by using the `ALTER SYSTEM SET cursor_sharing=FORCE` command.
- Monitor database to see if parse activity is reduced and library cache reloads shrink.
- If using an `SPFILE`, save the new configuration values so Oracle reuses them each time the database is stopped and re-started. If using an `Init.ora` file, add the `cursor_sharing=FORCE` parameter to the file.

Dictionary Cache Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

Oracle's shared pool offers an number of tuning possibilities and is made up of two main memory areas:

- 1 Library Cache
- 2 Data Dictionary Cache

The dictionary cache hit ratio offers another key indicator in determining the performance of the shared pool. It shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much of this vital reference information be kept in RAM as possible.

NOTE: This statistic is also available on the [Memory home page](#).

Metrics

Just as with the library cache, a high data dictionary cache hit ratio is desirable. Strive for a hit ratio between 90-100%, with 95% being a good performance benchmark.

NOTE: When a database is first started, the data dictionary cache hit ratio is not at an optimal level because all references to object definitions are relatively new, and as such, must be placed into the shared pool. Look for hit ratios in the eighty's for new database startups. If, however, after a solid hour or two of steady database time, the data dictionary cache hit ratio has not increased to desirable levels, increase the `shared_pool_size` parameter.

Databases supporting applications that involve large number of objects (such as an Oracle Financials installation) should have larger than normal shared pools to support the required object definitions.

Although each parameter is not individually tunable (it was in Oracle6), you can see which area of the dictionary cache could be pulling the overall hit ratio down.

Troubleshooting

If a problem is found in Oracle8i or earlier, do the following:

- Edit the `Init.ora` file for the database.
- Increase the amount of `shared_pool_size` to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

If a problem is found in Oracle9i or later, do the following:

- Increase the size of the `shared_pool_size` parameter through use of the `ALTER SYSTEM SET shared_pool_size` command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an `SPFILE`, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

Memory Sort Ratio

- [Metrics](#)
- [Troubleshooting](#)

Oracle's SGA is not the only memory structure used by Oracle for database work. One of the other memory areas used by Oracle8i and earlier for normal activity is an area set aside for sort actions. When a sort operation occurs, Oracle attempts to perform the sort in a memory space that exists at the operating system level. If the sort is too large to be contained within this space, it continues the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE. Oracle records the overall number of sorts that are satisfied in memory as well as those that end up being finalized on disk. Using these numbers, you can calculate the percentage of memory sorts vs. disk sorts and get a feel for how fast your sort activity is being resolved. Obviously, memory sorts completes many times faster than sorts forced to use physical I/O to accomplish the task at hand.

Oracle9i or later now has the option of running automatic PGA memory management. Oracle has introduced a new Oracle parameter called `pga_aggregate_target`. When the `pga_aggregate_target` parameter is set and you are using dedicated Oracle connections, Oracle ignores all of the PGA parameters in the Oracle file, including `sort_area_size`, `hash_area_size` and `sort_area_retained_size`. Oracle recommends that the value of `pga_aggregate_target` be set to the amount of remaining memory (less a 10% overhead for other server tasks) on a server after the instance has been started.

NOTE: This statistic is available on the Home page, Memory home page, and the Users home page.

Metrics

If the memory sort ratio falls below 90%, and you are on Oracle8i or earlier, increase the parameters devoted to memory sorts - `sort_area_size` and `sort_area_retained_size`.

For Oracle9i or later, investigate the use of `pga_aggregate_target`. Once the `pga_aggregate_target` has been set, Oracle automatically manages PGA memory allocation, based on the individual needs of each Oracle connection. Oracle9i or later allows the `pga_aggregate_target` parameter to be modified at the instance level with the `alter system` command, thereby lets you dynamically adjust the total RAM region available to Oracle9i.

Oracle9i also introduced a new parameter called `workarea_size_policy`. When this parameter is set to `automatic`, all Oracle connections benefits from the shared PGA memory. When `workarea_size_policy` is set to `manual`, connections allocates memory according to the values for the `sort_area_size` parameter. Under the `automatic` mode, Oracle tries to maximize the number of work areas that are using optimal memory and uses one-pass memory for the others.

Troubleshooting

If you find a problem, do the following:

- Edit the `Init.ora` or `SPFILE` file for the database.
- Increase the amount of `sort_area_size` to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value. EVERY user receives this amount for sorting.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

In addition to increasing the amount of memory devoted to sorting, find inefficient SQL that cause needless sorts. For example, `UNION ALL` does not cause a sort whereas `UNION` does in an SQL query (to eliminate duplicate rows). `DISTINCT` is frequently misapplied to SQL statements and causes unnecessary sort actions.

There are times you simply cannot stop sort activity. This being the case, try to keep it in memory whenever possible. However, large data warehousing systems oftentimes simply exhaust RAM sort allotments, so if disk sorts must occur, try to ensure three things:

- 1 Your user's TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.

NOTE: For Oracle9i or later, you can specify a default tablespace other than SYSTEM for every user account that is created.
- 2 The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.
- 3 The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.

Parse/Execute Ratio

Each time a new SQL statement is submitted to Oracle, the kernel must 'parse' the statement, which involves syntax checks, security checks, and object validations. The Parse/Execute Ratio shows the percentage of SQL executed that did not incur a hard parse.

NOTE: This statistic is available on the Home page, Memory home page, and the Users home page.

Metrics

Seeing low values might indicate that users are firing SQL with many hard-coded literals instead of using bind variables within an application. High values (90% and greater) generally indicate Oracle is saving precious CPU by avoiding heavy parse tasks.

Troubleshooting

The best way to reduce unnecessary parse activity is to encourage SQL statement reuse. This can be done by promoting SQL execution through the use of stored procedures or applications where bind variables can be used. Oftentimes, literals in otherwise identical SQL statements can cause unneeded parse work for Oracle. The use of bind variables can counter that problem.

If you determine that SQL literals are causing SQL to not be reused, do the following (in Oracle 8.1.6 and later):

- Change the cursor_sharing parameter to FORCE by using the ALTER SYSTEM SET cursor_sharing=FORCE command.
- Monitor database to see if parse activity is reduced and library cache reloads shrink.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started. If using an Init.ora file, add the cursor_sharing=FORCE parameter to the file.

Buffer Busy Waits

Buffer busy waits occur when a process needs to access a data block in the buffer cache, but cannot because it is being used by another process. So it must wait. A wait event generally happens because a buffer is being read into the buffer cache by another process or the buffer is in the buffer cache, but cannot be switched to a compatible mode immediately.

Metrics

Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables.

Troubleshooting

On Oracle8i or earlier, the remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.

For Oracle9i or later, you can use the automatic segment management feature in Oracle9i locally-managed tablespaces to help make free list problems a thing of the past. Using an UNDO tablespace in 9i or later can help remedy any rollback contention problem.

You can also obtain which objects have actually experienced buffer busy waits in Oracle9i or later by querying the `sys.v_$segment_statistics`. This view is not populated unless the configuration parameter `statistics_level` is set to TYPICAL or ALL.

Rollback Contention Ratio

Rollback segments are used by Oracle to hold data needed to rollback (or undo) any changes made through inserts, updates, or deletes to various Oracle objects. They also allow Oracle to have read consistency for long running queries, are used for recovery purposes, and play a role during exports of database information. In a heavy transaction processing environment, rollback segments are accessed continuously and therefore are subject to contention problems. The Rollback Contention Ratio helps identify contention occurring on the system relating to rollbacks.

Metrics

Overall, if the rollback contention ratio approaches 1% or more, create more rollback segments. Also consider creating a specialized, larger, rollback segment to be used by long running transactions. Doing so alleviates dynamic rollback extensions and cuts down heavily on ORA-01555 Snapshot Too Old errors.

Troubleshooting

Begin by creating new rollback segments and altering them to be online for use. Then monitor the overall contention ratio to see if it begins to drop.

If you are using Oracle9i or later, consider using an UNDO tablespace and allowing Oracle to automatically control rollback segment management.

Latch Miss Ratio

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism - so either you get it or you do not and are forced to continually retry.

The latch miss ratio defines the number of times a process obtained a willing-to-wait latch vs. missing the attempt.

Metrics

If the latch miss ratio exceeds 1%, you should take action to resolve the amount of latch contention.

Troubleshooting

Examine the details regarding the latch contention. Increasing the `shared_pool_size` can assist in latch problems also. The table below describes latches:

Latch	Description
Cache buffer chain latch	Protects paths to database block buffers in the buffer cache. High I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the <code>db_block_buffers</code> or <code>db_cache_size</code> parameter) or by adding more LRU latch chain latches with the <code>db_block_lru_latches</code> parameter.
Library cache latches	Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.

Other routine latch contention problems used to include the redo allocation and redo copy latches, but these have pretty much been made obsolete in Oracle 8.1.5 and later.

Parallel Query Busy Ratio

Oracle's parallel query feature, when used properly, allows for terrific increases in performance for many SQL and utility operations. Parallel operations can be introduced through SQL hints, specified in an object's DDL, or used in command line arguments for utility programs (like SQL*Loader). To effectively service parallel query requests, ensure that enough query servers exist in the database instance. The Parallel Query Busy Ratio is an indicator of how busy all the servers are on the database in question.

Metrics

If the Parallel Query Busy Ratio approaches 80-90%, add more query servers to the database, or examining parallel requests to ensure they are being used in an efficient and necessary manner.

Troubleshooting

To fix, do the following:

- Edit the `Init.ora` file or `SPFILE` for the database.
- Increase the amount of `parallel_max_servers` to a higher value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

You can also investigate the use of the `parallel_automatic_tuning` parameter in Oracle 8.1 and later.

Free Shared Pool Percent

Oracle's shared pool need not use all of the memory given to it through the `shared_pool_size` parameter. If the database does not have many object and code definitions to reference, then the shared pool can contain an abundance of free memory that is not being used.

Metrics

Under-allocating the shared pool size can have a serious impact on your database's performance, but over-allocating the shared pool can have run time ramifications as well. If you have a good chunk of memory allocated to the Oracle shared pool that is never used, it might be more of a performance enhancement to reduce the shared pool amount and instead give the memory to the buffer/data cache, or even back to the operating system itself. In terms of knowing when to reduce the shared pool, a good benchmark is continually seeing 2-3MB of free memory.

On the other hand, if after an hour or so of beginning database operation, you see that virtually no free memory is left in the shared pool, or you are seeing ORA-4031 errors (that indicate definitions cannot find enough contiguous free space in the shared pool), increase the pool by 10% or more.

Troubleshooting

If you continuously see little or no free memory in the shared pool, do the following:

For Oracle8i or earlier:

- Edit the Init.ora file for the database.
- Increase the amount of `shared_pool_size` to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

For Oracle9i or later:

- Increase the size of the `shared_pool_size` parameter through use of the `ALTER SYSTEM SET shared_pool_size` command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

You can also use the `ALTER SYSTEM FLUSHED SHARED_POOL` command to remove all objects from the shared pool and start with a clean slate.

Problem Tablespaces

The Problem Tablespaces metric is a count of all tablespaces in the database whose free space percentage amount has fallen below a user-defined limit.

Metrics

A rule of thumb for dynamic tablespaces (those with growing objects) is to keep a minimum of 10-15% free space available for object growth.

Troubleshooting

There are two ways to prevent a tablespace from running out of available free space:

- Turn `AUTOEXTEND` on for the underlying tablespace's datafiles. This allows them to automatically grow when free space in the datafile has been exhausted.
- Using the `ALTER TABLESPACE ... ADD DATAFILE...` command, you can manually add a new datafile to a tablespace that is about to run out of available free space.

Problem Objects

The Problem Objects statistic is a count of all objects in the database that are in danger of reaching their maximum extent limit or cannot allocate their next extent of free space because of a lack of overall or contiguous free space in their parent tablespace.

Metrics

Modify any object approaching their maximum extent limit or unable to allocate a new extent of space so they can continue to grow in size.

Troubleshooting

Depending on the situation, there are a number of things you can do to prevent object extent problems:

- Turn AUTOEXTEND on for the underlying parent tablespace's datafiles. This allows a tablespace to automatically grow when free space in the datafile has been exhausted, and allows an object to extend even when little or no current free space is available.
- Using the ALTER TABLESPACE ... ADD DATAFILE... command, you can manually add a new datafile to a tablespace that is about to run out of available free space.
- You can alter an object that is at or near their maximum extent limit so that the object has unlimited extents.
- With Oracle 8.1.5 and later, you can use locally-managed tablespaces to ensure that no object ever reaches its maximum extent limit, because all objects are allowed unlimited extents.
- An object can be reorganized into another tablespace or reorganized in general to reduce the number of extents the object currently takes up.

Top System Bottlenecks

When viewing wait statistics, there are many levels of detail that you can view. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. Viewing these numbers can help you determine which wait events have caused the most commotion in a database thus far. The Top System Bottlenecks section identifies the top waits that have occurred on the Oracle database based on the number of waits per event.

Metrics

None.

Troubleshooting

Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics. For example, a 'db file scattered read' event is typically indicative of table scan operations. If you see many of these events, then you can begin to see if large table scans are occurring in the database. Like the 'db file scattered read' event, each wait event has its own meaning and individual end-resolution diagnosis.

After looking at system-level wait activity, you can discover which current connections are responsible for waits at the system level. Performance Analyst reports on historical and current wait events at the session level, making this investigation easy to accomplish.

Top Session Bottlenecks

When viewing wait statistics, there are many levels of detail that you can view. The first level is the system view, which provides a global, cumulative snapshot of all the waits that have occurred on a system. The second level is a historical look at waits from a session level. The third is which sessions are currently experiencing waits. The Top Session Bottlenecks section identifies the top sessions that are currently waiting, based on their wait time in seconds.

Metrics

None.

Troubleshooting

Appendix A in the Oracle Reference manual contains a listing and description of every current wait event defined in Oracle. DBAs unfamiliar with what each event represents should keep this listing close by as they examine wait-based event metrics.

The most common wait viewed at the session level is an 'enqueue' wait, which typically identifies lock contention. If enqueue waits are observed, then you can check the "Current Object Blocks" count on the Performance Analyst Home page, as well as the [Users page](#) which displays locks and blocking locks detail.

As with an enqueue event, each wait event has its own meaning and individual end-resolution diagnosis.

Current Object Blocks

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up. Blocks are most often caused by user processes holding exclusive locks and not releasing them via a proper COMMIT frequency.

NOTE: This statistic is also called Sessions Blocked on the [Users home page](#) and Session Blocks on the [Objects home page](#).

Metrics

Investigate any indicator above zero immediately before the situation has a chance to mushroom.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied. You can issue a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Analyst, but preventing the blocking lock situation in the first place is where it gets tricky. You can look at the [Users Detail](#) and view all current blocking locks to see exactly which sessions are holding the currently restrictive locks.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT ... FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Enqueue Waits

An enqueue is an advanced locking device that allows multiple database processes to share certain resources. Enqueue waits typically occur when sessions wait to be granted a requested lock. Sometimes these locks are internal Oracle locks while other times they could be locks for rows of data in a table. Enqueues are issued implicitly by Oracle.

NOTE: This statistic is available on the Home page and the [Objects home page](#).

Metrics

Investigate any enqueue waits that read consistently above one or more (delta statistics).

Troubleshooting

Removing contention for enqueues is almost always an application design issue. If many enqueue waits are seen, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. Examine the use of indexes to make sure all referencing foreign keys are indexes and that SQL is tuned to not tarry over rows in the database during modification operations.

Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).

Free List Waits

Free lists are lists of Oracle data blocks that contain free space for an Oracle object. Every table has at least one free list. Free lists are used to locate free blocks of space when a request is made of a table for the insertion of a row. Free list contention can reduce the performance of applications when many processes are involved in the insertion of data to the same table.

NOTE: This statistic is available on the Home page and the [Objects home page](#).

Metrics

If consistent numbers for free list waits continues to appear, add additional free lists to the most dynamic database objects through the use of the STORAGE parameter. Another indicator of insufficient free lists are consistent, non-zero numbers for the buffer busy wait event.

Troubleshooting

If free list waits are observed, add more free lists to tables and indexes with high insert rates. For Oracle9i or later, objects can be created inside of locally-managed tablespaces that use the automatic segment management feature, which entirely does away with free lists.

Total Used Space/Total Free Space

These statistics represent the total used and free space available in all tablespaces/datafiles in the database. Although good to know, a more detailed listing by tablespace is needed to determine where any actual space shortages exist in the database.

NOTE: These statistics are in the Storage Analysis section of the Performance Analyst Home page and the Space home page.

Metrics

If any one tablespace begins to approach 90% used (and the objects contained within it are dynamic and growing as opposed to static), take action to prevent any future space allocation errors.

Troubleshooting

Here are a some things you can do to prevent a tablespace from running out of available free space:

- Turn AUTOEXTEND on for the underlying tablespace's datafiles. This allows them to automatically grow when free space in the datafile has been exhausted.
- Using the ALTER TABLESPACE ... ADD DATAFILE... command, you can manually add a new datafile to a tablespace that is about to run out of available free space.

For more information, see [Storage Analysis](#).

Archive Log

Oracle can be placed into archivelog mode, which tells the database to make copies of the online redo log files for point-in-time recovery purposes. The Archive Log statistic displays whether the database is running in archivelog mode or not. This information is displayed in the Storage Analysis section of the Performance Analyst Home page.

Metrics

None.

For more information, see [Storage Analysis](#).

Top Processes

When the database population as a whole experiences a system slowdown, it is not uncommon to find one or two users who are responsible for bringing the system to its knees. In the best of worlds, users have an evenly divided amount of memory usage, disk I/O, CPU utilization, and parse activity. Unfortunately, this is not usually the case. Users submit large batch jobs during peak OLTP activity, or when sessions that are firing off untuned queries on a critical system.

If you are seeing a slowdown in your database, and cannot seem to find a root cause, one thing to examine is the resource consumption of the leading sessions on a system. A handful of users can overwhelm the I/O capabilities of Oracle (through untuned queries or runaway batch jobs) or hammer the CPU or memory structures of the database and server.

Performance Analyst makes it easy to pinpoint the top sessions by showing the leading processes at the physical I/O, logical I/O, memory, and CPU usage levels.

Metrics

If any one session uses more than 50% of a total resource (CPU, memory, etc.) go into the session to find out what they are currently executing.

Active User Processes

The Active User Processes statistic is the total number of active and open threads reported in the database. Active Sessions displays the number of processes actively performing work.

Metrics

None.

For more information, see [Inactive User Processes](#).

Inactive User Processes

The Inactive User Processes statistic is the total number of threads logged on to the database that are idle at the current time.

Metrics

A large number of inactive users could indicate user sessions that have mistakenly been left logged on. Because each user thread consumes a portion of memory on the Oracle server, sever any sessions not needing a connection to reduce resource usage.

Troubleshooting

Double-click this statistic to open the [Top Sessions tab](#) of the Users Detail page. On this tab you can check sessions that have many seconds idle and/or that have been logged on for very long periods of time as indicated by the logon time column. After verifying that the session is no longer necessary, you can KILL the session.

For more information, see [Active User Processes](#).

Memory Page Statistics

The Memory home page includes the following sections:

- [Bottleneck Analysis](#)
- [Key Ratio Analysis](#)
- [SGA Analysis](#)
- [SQL Analysis](#)
- [Workload Analysis - Top Memory Hogs](#)
- [Buffer Cache Hit Ratio](#)
- [Buffer Pool Allocations](#)
- [Buffer Pool Usage](#)
- [Data Dictionary Cache Hit Ratio](#)
- [Free Memory in Shared Pool](#)
- [Library Cache Hit Ratio](#)
- [Memory Sort Ratio](#)
- [Disk Sort Ratio](#)
- [Session Leaders - Memory](#)
- [SGA](#)

Related Topics

[Memory Detail](#)

[Home Page Statistics](#)

[I/O Page Statistics](#)

[Objects Page Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

[Users Page Statistics](#)

[Network Statistics](#)

[Contention Statistics](#)

Key Ratio Analysis - Memory

Object-related database activity can be examined using both ratio-based and wait/bottleneck-based analysis. Ratio-based analysis involves examining a number of key database ratios and statistical readings that can be used to indicate how active certain object types are. Performance ratios serve as roll-up mechanisms for busy DBAs to use for at-a-glance performance analysis.

When using ratio-based analysis, there are some standards to adhere to. To start with, many of the formulas that make up ratio-based analysis must be derived from delta measurements instead of cumulative statistics. Many of the global ratios that you examines come from the v\$sysstat performance view. The performance view maintains a count of all the occurrences (in the VALUE column) of a particular database incident (in the NAME column) since the database was brought up. For databases that are kept up for long periods of time, these values can grow quite large and impacts how a particular ratio is interpreted. However, if delta statistics are used (taking, for a specified sampling period, the before and after counts of each statistic that make up a ratio), then an accurate and current portrayal of the various ratios can be had.

A final thing to remember about using ratio-based analysis is that, while there are a number of rules of thumb that can be used as starting points to begin the evaluation of database performance, DBAs must determine each database's individual 'personality' with respect to the key performance ratios. Some hard and fast rules simply do not apply to every database. The danger in using blanket ratio standards is that they can lead you to haphazardly take action, which can at times contribute nothing to the situation, and sometimes even degrade performance.

The following memory ratios are used on the Performance Analyst Memory home page to succinctly communicate the general overall memory performance levels of the monitored database:

- [Buffer Cache Hit Ratio](#)
- [Library Cache Hit Ratio](#)
- [Dictionary Cache Hit Ratio](#)
- [Memory Sort Ratio](#)
- [Parse/Execute Ratio](#)
- [Free Shared Pool Percent](#)

For more information, see [Memory Page Statistics](#).

Bottleneck Analysis - Memory

When an Oracle database is up and running, every connected process is either busy doing work or waiting to perform work. A process that is waiting may mean nothing in the overall scheme of things or it can be an indicator that a database bottleneck exists. You can use Bottleneck Analysis to determine if perceived bottlenecks in a database are contributing to a performance problem.

Memory bottlenecks can definitely cause performance degradation in an otherwise well-running database. Typically, these bottlenecks center around Oracle's buffer/data cache, library cache, and occasionally log buffer memory regions. To help you identify such problems, the following statistics are presented on the Memory home page:

- [Buffer Busy Waits](#)
- [Free Buffer Wait Average](#)
- [Object Reloads](#)
- [Redo Log Space Wait Time](#)
- [Top Latch Misses](#)

For more information, see:

[Memory Home Page](#)

[Buffer Busy Ratio](#)

SQL Analysis - Memory

Much of a database's overall performance can be attributed to SQL statement execution. Poorly optimized SQL statements can drag an otherwise well-configured database down in terms of end user response times. SQL statements that use much memory can also cause a problem in a database. The SQL Analysis for memory shows what SQL statements have consumed the largest percentages of shareable, persistent, and runtime memory

Before you can identify problem SQL in your database, you have to ask the question, "What is bad SQL?" What criteria do you use when you begin the hunt for problem SQL in your critical systems? Understand that even the seasoned experts disagree on what constitutes efficient and inefficient SQL; so there is no way to sufficiently answer this question to every Oracle professional's satisfaction. The table lists some general criteria you can use when evaluating the output from various database monitors or personal diagnostic scripts:

Criteria	Description
Overall Response (Elapsed) Time	The time the query took to parse, execute, and fetch the data needed to satisfy the query. It should not include the network time needed to make the round trip from the requesting client workstation to the database server. This statistic is available in Oracle9i or later.
CPU Time	The CPU time the query took to parse, execute, and fetch the data needed to satisfy the query.
Physical I/O	This is often used as the major statistic in terms of identifying good vs. bad SQL, this is a measure of how many disk reads the query caused to satisfy the user's request. While you certainly want to control disk I/O where possible, it is important that you not focus solely on physical I/O as the single benchmark of inefficient SQL. Make no mistake, disk access is slower than memory access and also consumes processing time making the physical to logical transition, but you need to look at the entire I/O picture of a SQL statement, which includes looking at a statements' logical I/O as well.
Logical I/O	The number of memory reads the query took to satisfy the user's request. The goal of tuning I/O for a query should be to examine both logical and physical I/O, and use appropriate mechanisms to keep both to a minimum.
Repetition	The number of times the query has been executed. A problem in this area is not as easy to spot as the others unless you know your application well. A query that takes a fraction of a second to execute can be a headache on your system if it has executed erroneously (for example, a query that executes in a runaway PL/SQL loop) over and over again.

There are other criteria that you can examine like sort activity or access plan statistics (that show items like Cartesian joins and the like), but more often than not, these measures are reflected in the criteria listed above.

Fortunately, Oracle records all the above measures (some only in 9i), which makes tracking the SQL that has been submitted against an Oracle database much easier.

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you want answered:

- 1 What HAS been the worst SQL that has historically been run in my database?
- 2 What IS the worst SQL that is running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It could be that the query is in an inefficient PL/SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

There is the possibility that the SQL statement just is not tuned well. You can view Performance Analyst's Top SQL view and, if you have Embarcadero SQL Tuner installed, you can port the SQL over to SQL Tuner to better optimize the statement.

SGA Analysis - Memory

Most DBAs know all about the Oracle System Global Area (SGA). The SGA is Oracle's memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database. The table below describes Oracle memory structures:

Memory Structure	Description
Default buffer cache	Maintains data blocks when they are read from the database. If you do not specifically place objects in another data cache, then any data requested by clients from the database is placed into this cache. The memory area is controlled by the <code>db_block_buffers</code> parameter in Oracle8i and earlier and <code>db_cache_size</code> in Oracle9i or later.
Keep buffer cache	For Oracle 8 or later, you can assign various objects to a special cache that retains those object's requested blocks in RAM for as long as the database remains up. The keep cache's main is for often-referenced lookup tables that should be kept in memory at all times for fast access. The <code>buffer_pool_keep</code> parameter controls the size of this cache in Oracle8, while the <code>db_keep_cache_size</code> parameter handles the cache in Oracle9i or later. The keep pool is a sub-pool of the default buffer cache.
Recycle buffer cache	The opposite of the keep cache. When large table scans occur, the data that fills a memory cache will likely not be needed again and should be quickly discarded from RAM so that they do not occupy memory space and prevent needed blocks from assuming their place. Objects containing such data can be assigned to the recycle pool to ensure that such a thing does indeed occur. The <code>buffer_pool_recycle</code> parameter controls the size of this cache in Oracle8 and earlier, while the <code>db_recycle_cache_size</code> parameter handles the cache in Oracle9i or later.
Specific block size caches	For Oracle 8 or later, you can create tablespaces whose blocksize differs from the overall database blocksize. When data is read into the SGA from these tablespaces, their data has to be placed into memory regions that can accommodate their special block size. Oracle9i or later has memory settings for 2K, 4K, 8K, 16K, and 32K caches. The configuration parameter names are in the pattern of <code>db_nk_cache_size</code> .
Shared pool	Holds object structure as well as code definitions, and other metadata. Setting the proper amount of memory in the shared pool assists a great deal in improving overall performance with respect to code execution and object references. The <code>shared_pool_size</code> parameter controls the memory region.

Memory Structure	Description
Large pool	For Oracle 8 or later, you can configure an optional, specialized memory region called the large pool that holds items for shared server operations, backup and restore tasks, and other miscellaneous things. The <code>large_pool_size</code> parameter controls the memory region. The large pool is also used for sorting when the multi-threaded server (MTS) is implemented.
Java pool	Handles the memory for Java methods, class definitions, etc. The <code>java_pool_size</code> parameter controls the amount of memory for this area.
Redo log buffer	Buffers modifications that are made to the database before they are physically written to the redo log files. The <code>log_buffer</code> configuration parameter controls the memory area.

Oracle also maintains a fixed area in the SGA that contains a number of atomic variables, pointers, and other miscellaneous structures that reference areas of the SGA.

For more information, see [Memory Home Page](#).

Workload Analysis - Top Memory Hogs

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Oftentimes, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system level (ORA-4030 errors).

If your database server does not have an overabundance of memory, periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 25-50% of the total memory usage, investigate the sessions further to see the activities they are performing.

For more information, see [Memory Home Page](#).

Buffer Cache Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

The buffer cache hit ratio is an indicator of how often user requests for data are satisfied through memory vs. being physically read from disk. Data read from memory produces user response times many times faster than when that same data is read from disk. Keeping physical I/Os to an absolute minimum is one of the Oracle buffer cache's purposes in life.

The table below describes the key counters Performance Center uses to calculate the buffer cache hit ratio:

Key Counter	Description
DB BLOCK GETS	The number of times a CURRENT block was requested. You review values for this statistic in two places. First, go to Memory Detail>Session Detail>Session I/O. Second, you can add this statistic to the Trends view.
CONSISTENT GETS	Data read from rollback segments in memory.
PHYSICAL READS	Data read physically from disk.
Direct Reads	Data read physically from disk that bypasses the buffer cache. Direct reads are filtered out of overall physical reads so an accurate cache hit ratio can be determined.

Dividing the data read from memory by data read from disk yields the cache hit ratio.

TIP: Click this statistic to drill down to the [Hit Ratio by User tab](#) of the Memory Detail view.

Metrics

To help ensure excellent performance, you want to keep your cache hit ratio in the neighborhood of 90% or higher. However, every database has its own 'personality' and can exhibit excellent performance with below average readings for the cache hit ratio. Excessive logical I/O activity can produce a very high cache hit ratio while actually degrading overall database performance.

You should investigate consistent low readings of 60% or lower.

NOTE: You must cycle the Oracle server to allow any increases in `db_block_buffers` to take effect.

NOTE: For Oracle8i or lower, the adjustment of the `db_block_buffers` tuning parameter is required. For Oracle9i or higher, the `db_cache_size` parameter is the parameter that needs attention. Any increases in `db_block_buffers` to take effect on Oracle8i or lower, the database must be cycled. The `db_cache_size` parameter in Oracle9i or later, however, is dynamic and can be altered without stopping and starting the database instance.

To view individual session hit ratios that can be depressing the overall cache hit ratio for the database, drill down into the overall buffer cache hit ratio.

Troubleshooting

If a problem is found in Oracle8i or earlier:

- Edit the `Init.ora` file for the database.
- Increase the amount of `db_block_buffers` to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later:

- Increase the size of the `db_cache_size` parameter through use of the `ALTER SYSTEM SET db_cache_size` command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an `SPFILE`, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

Buffer Pool Usage

- [Metrics](#)
- [Troubleshooting](#)

Buffer cache usage statistics show the state and use of the buffers currently in the buffer cache and they include a count of how many there are for each state. Typically, the statuses of the buffers are:

- Free
- Read and modified

- Read and not modified
- Currently being read

Metrics

This grouping of statistics is quite helpful in determining if you have an overabundance of block buffers allocated. Consistently seeing large number of free buffers clues you in to the fact that you can reduce the amount of block buffers in the cache and give memory back to the operating system.

If, however, you see no free buffers within the first hour of bringing your Oracle database up, you can consider adding more buffers to the cache.

Troubleshooting

If you find a problem, do the following:

- 1 Edit the Init.ora file for the database.
- 2 Increase the amount of db_block_buffers to a higher value if free buffers are not found (take care not to over-allocate; ensure enough free memory exists on server before increasing value). Reduce the number of an overabundance of FREE buffers are present.
- 3 Cycle the Oracle server when possible to allow the new value to take effect.
- 4 Monitor the new value to see if more numbers of free buffers show up for the FREE status.

Buffer Pool Allocations

- [Metrics](#)
- [Troubleshooting](#)

Because you can reference data objects with different usage patterns, Oracle8 offers the option to intelligently place objects into one of three buffer caches:

Buffer Cache	Description
KEEP	Minimizes misses in the buffer cache. Small, frequently referenced objects are candidates for the KEEP buffer pool.
RECYCLE	Lets you avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM. The RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.
DEFAULT	The traditional cache for all other data objects.

NOTE: Unless you specify the KEEP or RECYCLE buffer cache, Oracle automatically places objects into the DEFAULT buffer cache.

TIP: Click this statistic to drill down to the [Buffer Pool tab](#) of the Memory Detail view.

Metrics

When looking at overall database I/O activity, you should keep an eye out for objects that can lend themselves to placement into a particular buffer cache. Consider using a KEEP cache for relatively small, frequently accessed tables that require fast response times. Large tables with random I/O activity and are scanned from beginning to end a lot are good candidates for a RECYCLE cache.

Troubleshooting

You can use the `STORAGE...BUFFER_POOL` option to place objects into different buffer pools at object creation time. You use the `ALTER` command to set existing objects different pool.

TIP: Tables, partitions, and indexes can be placed into the different caches.

If you just want to use the `DEFAULT` buffer pool and not enable any special caches, you can still encourage Oracle to keep certain objects in the buffer cache as long as possible by using the `CACHE` parameter. For example, issuing the command `ALTER TABLE...CACHE` specifies that the blocks retrieved for this table be placed at the most recently used end of the LRU list in the `DEFAULT` buffer cache when a full table scan is performed. The `CACHE` hint can also be used in SQL statements to cache a table, but used in this form, Oracle only caches the blocks until the next time the database is shut down. Once the database comes back up, the `CACHE` hint would have to be issued in an SQL statement again to cache the needed blocks of data.

Free Memory and Used Memory in Shared Pool

- [Metrics](#)
- [Troubleshooting](#)

Oracle's shared pool does not need to use all of the memory provided through the `shared_pool_size` parameter. If the database does not have many object and code definitions to reference, the shared pool can contain an abundance of unused free memory. This statistic is tracked by Oracle in its performance views.

TIP: Click this statistic to drill down to the [Shared Pool tab](#) of the Memory Detail view.

Metrics

Under-allocating the shared pool size can have a serious impact on your database's performance and over-allocating the shared pool can have runtime ramifications. If you have a good chunk of memory allocated to the Oracle shared pool that is never used, it might be more of a performance enhancement to reduce the shared pool amount and instead give the memory to the buffer cache, or even back to the operating system itself. In terms of knowing when to reduce the shared pool, a good benchmark is continually seeing 2-3 megabytes (MB) of free memory.

On the other hand, if after an hour or so of beginning database operation, you see virtually no free memory is left in the shared pool, or you are seeing `ORA-4031` errors (that indicate definitions cannot find enough contiguous free space in the shared pool), you should definitely look into increasing the pool by 10% or more.

Troubleshooting

If you continuously see little or no free memory in the shared pool then you can do the following on Oracle8i or lower:

- Edit the `init.ora` file for the database.
- Increase the amount of `shared_pool_size` to a higher value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value.)
- Cycle the Oracle server when possible to allow the new value to take affect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later:

- Increase the size of the `shared_pool_size` parameter through use of the `ALTER SYSTEM SET shared_pool_size` command value (take caution to not over-allocate; ensure enough free memory exists on server before increasing value.)
- Monitor the new value to see if performance improves.

- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

You can also use the `ALTER SYSTEM FLUSHED SHARED_POOL` command to remove all objects from the shared pool and start with a clean slate.

Library Cache Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

Oracle's shared pool offers a number of tuning possibilities and is made up of two main memory areas:

- 1 Library Cache
- 2 Data Dictionary Cache

The library cache hit ratio offers a key indicator in determining the performance of the shared pool. It holds commonly used SQL statements - basically database code objects. The library cache hit ratio shows how often SQL code is reused by other database users vs. the number of times an SQL statement is broken down, parsed, and then loaded (or reloaded) into the shared pool.

You can improve performance by encouraging the reuse of SQL statements so expensive parse operations can be avoided. The library cache assists this tuning effort.

NOTE: This statistic is also available on the Home page, the Memory home page and on the Library Cache tab of the Memory Detail.

TIP: Click this statistic to drill down to the [Library Cache tab](#) of the Memory Detail view.

Metrics

A high library cache hit ratio is a desirable thing. Strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse.

NOTE: When a database is first started, the library cache hit ratio is not at an optimal level because all code being used is relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, increase the `shared_pool_size` parameter.

Other red flags that can indicate a too small shared pool include:

- A wait count for the event 'latch free' of 10 or greater.
- The library cache wait count of two or greater.

You can track these indicators in Performance Center's Contention performance category view.

A way of improving the library cache hit ratio is by encouraging code reuse through the implementation of bind variables. Discouraging hard coded literals in application code and instead making use of variables bound at run time aids in the reuse of SQL code that is maintained in Oracle's shared pool.

NOTE: Bind variables can have an affect on the cost-based optimizer though.

A second way is to pin frequently used code objects in memory so they are available when needed, by using the system supplied `DBMS_SHARED_POOL` package. You can use Performance Center to view objects in the shared pool that are always present and/or have increasing reload numbers to help identify objects that are good candidates for pinning.

You can use Performance Center to view objects in the shared pool that are always present and/or have increasing reload numbers to help identify objects that are good candidates for pinning.

Troubleshooting

If a problem is found in Oracle8i or earlier:

- Edit the Init.ora file for the database.
- Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later:

- Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

Dictionary Cache Hit Ratio

- [Metrics](#)
- [Troubleshooting](#)

Oracle's shared pool offers a number of tuning possibilities and is made up of two main memory areas:

- 1 Library Cache
- 2 Data Dictionary Cache

The dictionary cache hit ratio offers another key indicator in determining the performance of the shared pool. It shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much of this vital reference information be kept in RAM as possible.

TIP: Click this statistic to drill down to the [Data Dictionary tab](#) of the Memory Detail view.

Metrics

Just as with the library cache, a high data dictionary cache hit ratio is desirable. Strive for a hit ratio between 90-100%, with 95% being a good performance benchmark.

NOTE: When a database is first started, the data dictionary cache hit ratio is not at an optimal level because all references to object definitions are relatively new, and as such, must be placed into the shared pool. Look for hit ratios in the eighty's for new database startups. If, however, after a solid hour or two of steady database time, the data dictionary cache hit ratio has not increased to desirable levels, increase the shared_pool_size parameter.

Databases supporting applications that involve large number of objects (such as an Oracle Financials installation) should have larger than normal shared pools to support the required object definitions.

Although each parameter is not individually tunable (it used to be in Oracle6), you can see which area of the dictionary cache could be pulling the overall hit ratio down.

Troubleshooting

If a problem is found in Oracle8i or earlier:

- Edit the Init.ora file for the database.
- Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

If a problem is found in Oracle9i or later:

- Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

Memory Sort Ratio

- [Metrics](#)
- [Troubleshooting](#)

Oracle's SGA is not the only memory structure used by Oracle for database work. One of the other memory areas used by Oracle8i and earlier for normal activity is an area set aside for sort actions. When a sort operation occurs, Oracle attempts to perform the sort in a memory space that exists at the operating system level. If the sort is too large to be contained within this space, it continues the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE. Oracle records the overall number of sorts that are satisfied in memory as well as those that end up being finalized on disk. Using these numbers, you can calculate the percentage of memory sorts vs. disk sorts and get a feel for how fast your sort activity is being resolved. Obviously, memory sorts completes many times faster than sorts forced to use physical I/O to accomplish the task at hand.

Oracle9i or later now has the option of running automatic PGA memory management. Oracle has introduced a new Oracle parameter called pga_aggregate_target. When the pga_aggregate_target parameter is set and you are using dedicated Oracle connections, Oracle ignores all of the PGA parameters in the Oracle file, including sort_area_size, hash_area_size and sort_area_retained_size. Oracle recommends that the value of pga_aggregate_target be set to the amount of remaining memory (less a 10% overhead for other server tasks) on a server after the instance has been started.

TIP: Double-click these statistics to drill down to the [Disk Sort Detail tab](#) of the Users Detail view.

Metrics

If your memory sort ratio falls below 90%, and you are on Oracle8i or earlier, increase the parameters devoted to memory sorts - sort_area_size and sort_area_retained_size.

For Oracle9i or later, investigate the use of pga_aggregate_target. Once the pga_aggregate_target has been set, Oracle automatically manages PGA memory allocation, based upon the individual needs of each Oracle connection. Oracle9i or later allows the pga_aggregate_target parameter to be modified at the instance level with the alter system command, thereby lets you dynamically adjust the total RAM region available to Oracle9i.

Oracle9i also introduces a new parameter called `workarea_size_policy`. When this parameter is set to automatic, all Oracle connections benefits from the shared PGA memory. When `workarea_size_policy` is set to manual, connections allocates memory according to the values for the `sort_area_size` parameter. Under the automatic mode, Oracle tries to maximize the number of work areas that are using optimal memory and uses one-pass memory for the others.

Troubleshooting

If you find a problem, do the following:

- Edit the `Init.ora` or `SPFILE` file for the database.
- Increase the amount of `sort_area_size` to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value. EVERY user receives this amount for sorting.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

In addition to increasing the amount of memory devoted to sorting, find inefficient SQL that cause needless sorts. For example, `UNION ALL` does not cause a sort whereas `UNION` does in an SQL query (to eliminate duplicate rows). `DISTINCT` oftentimes is misapplied to SQL statements and causes unnecessary sort actions.

There are times you simply cannot stop sort activity. This being the case, try to keep it in memory whenever possible. However, large data warehousing systems oftentimes simply exhaust RAM sort allotments, so if disk sorts must occur, try to ensure three things:

- Your user's `TEMPORARY TABLESPACE` assignment is not the `SYSTEM` tablespace, which is the default assignment.
 - NOTE:** For Oracle9i or later, you can specify a default tablespace other than `SYSTEM` for every user account that is created.
- The `TEMPORARY TABLESPACE` assigned to your users is placed on a fast disk.
- The `TEMPORARY TABLESPACE` has the tablespace parameter `TEMPORARY` assigned to it, which allows sort activity to be performed in a more efficient manner.

Free Memory and Used Memory in Shared Pool

- [Metrics](#)
- [Troubleshooting](#)

Oracle's shared pool need not use all of the memory given to it through the `shared_pool_size` parameter. If the database does not have many object and code definitions to reference, then the shared pool can contain an abundance of free memory that is not being used.

TIP: Click this statistic to drill down to the [Shared Pool tab](#) of the Memory Detail view.

Metrics

Under-allocating the shared pool size can have a serious impact on your database's performance, but over-allocating the shared pool can have run time ramifications as well. If you have a good chunk of memory allocated to the Oracle shared pool that is never used, it might be more of a performance enhancement to reduce the shared pool amount and instead give the memory to the buffer/data cache, or even back to the operating system itself. In terms of knowing when to reduce the shared pool, a good benchmark is continually seeing 2-3MB of free memory.

On the other hand, if after an hour or so of beginning database operation, you see that virtually no free memory is left in the shared pool, or you are seeing `ORA-4031` errors (that indicate definitions cannot find enough contiguous free space in the shared pool), increase the pool by 10% or more.

Troubleshooting

If you continuously see little or no free memory in the shared pool, you can do the following:

If a problem is found in Oracle8i or earlier:

- Edit the Init.ora file for the database.
- Increase the amount of shared_pool_size to a higher value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor the new value to see if performance improves.

If a problem is found in Oracle9i or later:

- Increase the size of the shared_pool_size parameter through use of the ALTER SYSTEM SET shared_pool_size command value. Take caution to not over-allocate; ensure enough free memory exists on server before increasing value.
- Monitor the new value to see if performance improves.
- If using an SPFILE, save the new configuration values so Oracle reuses them each time the database is stopped and re-started.

You can also use the ALTER SYSTEM FLUSHED SHARED_POOL command to remove all objects from the shared pool and start with a clean slate.

Session Leaders - Memory

- [Metrics](#)
- [Troubleshooting](#)

Frequently one or two users can cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Frequently, user connections can get out of hand with memory consumption and extreme cases have caused headaches at both the database and operating system level (ORA-4030 errors).

NOTE: This statistic displays on both the [Users performance category view](#) and the Memory performance category view.

TIP: Click this statistic to drill down to the [Leading Sessions tab](#) of the Memory Detail view.

Metrics

If your database server does not have an overabundance of memory, you should periodically check to see two things:

- 1 Who are your heavy memory users
- 2 The total percentage of memory each user consumes.

If you see one or two users who have more than 5-15% of the total memory usage, you should further investigate the sessions to see what activities they are performing.

Troubleshooting

You can use the Session Leaders - Memory statistic to find the users with the greatest allocations of overall memory. Then you can drill down into the details to discover the memory components that make up each user's session. Pay particular attention to the amounts of memory usage associated with these areas:

Area	Definition
PGA	The Program Global Area is a private memory area devoted to housing the global variables and data structures for a single Oracle process.
UGA	The User Global Area contains session-specific information regarding open cursors, state information for packages, database link information, and more. Note that when using Oracle's Multi-threaded Server (MTS), the UGA can be moved up into the SGA.
Memory Sorts	Memory sorts contain a count of how many memory sorts a session has performed.

SGA

The SGA (System Global Area) is Oracle's memory structural area devoted to facilitating the transfer of data and information between clients and the Oracle database. The table below describes its main tunable components:

Component	Definition
Database Buffers	Maintains data blocks that are read from the database. Properly sizing the buffer cache goes a long way in improving response time performance.
Var Size	Holds object structure as well as code definitions. Setting the proper amount of memory in the variable size also assists a great deal in improving overall performance with respect to code execution and object references.
Redo Log Buffer	Buffers modifications that are made to the database before they are physically written to the redo log files.
Fixed Size	Contains a number of atomic variables, pointers, and other miscellaneous structures that reference areas of the SGA.

Metrics

None.

Memory Detail

The following tabbed pages are available in the Memory Detail view:

- [Buffer Pool](#)
- [Data Dictionary](#)
- [Library Cache](#)
- [Leading Sessions](#)
- [Objects in Memory](#)
- [Sessions Overview](#)
- [Shared Pool](#)

Data Dictionary Tab

- [Metrics](#)

Oracle's data dictionary is a component of the shared pool that contains system elements necessary for the parsing of SQL statements, security resolution, object definitions, and more.

The overall data dictionary cache hit ratio provides a key indicator in determining the performance of the shared pool, and shows how often object definitions are found in memory vs. having to read them in from disk. Because Oracle references the data dictionary many times when an SQL statement is processed, it is imperative that as much as possible of this vital reference information be kept in RAM.

The Data Dictionary tab shows the individual elements of the data dictionary along with their associated hit ratios. In versions 6.x of Oracle, these individual elements could be tuned, but in versions 7.x and later, the only main method for tuning them involves the adjustment of the entire shared pool setting. Although not tunable from an individual parameter level, each displayed element gives insight into which area of the data dictionary is either adding to or detracting from overall performance.

The table below describes the information available on this tab:

Column	Description
Parameter	The name of the individual data dictionary element.
Usage	The number of cache entries that contain valid data.
Gets	The number of requests for this element. To see what's happening to Dictionary Cache Gets in a given datasource, you can add this statistic to your Trends view.
Get Misses	The number of requests resulting in a cache miss To see what's happening to Dictionary Cache Misses in a given datasource, you can add this statistic to your Trends view.
Hit Ratio	The ratio of cache hits versus misses of total requests. The maximum is 100%.
Scans	The number of scan requests.
Scan Misses	The number of times that a scan failed to find the needed data in the cache.
Scan Completes	The number of times the list was scanned completely.
Modifications	The number of insert, update, and delete actions.
Flushes	The number of disk flushes.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

An overall high data dictionary cache hit ratio is desirable, as are high hit ratios in each individual parameter. You should strive for a hit ratio between 90-100%, with 95% being a good performance benchmark.

NOTE: When a database is first started, the data dictionary cache hit ratio is not at an optimal level because all references to object definitions are relatively new, and as such, must be placed into the shared pool. Look for hit ratios between 80-90% for new database startups. If, however, after an hour or two of steady database time, the data dictionary cache hit ratio and individual hit ratios, have not increased to desirable levels, you should look into the possibility of increasing the `shared_pool_size` parameter.

NOTE: Note that databases supporting applications that involve large number of objects (such as an Oracle Financials installation) should have larger than normal shared pools to support the required object definitions.

Buffer Pool Tab

- [Metrics](#)

Because data objects can be referenced with different usage patterns, Oracle8 and later offers the option to intelligently place objects into one of three buffer caches. The table below describes the types of buffer caches:

Cache	Description
KEEP	Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.
RECYCLE	Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.
DEFAULT	The traditional cache for all other data objects.

NOTE: Unless you specify the KEEP or RECYCLE buffer cache, Oracle automatically places objects into the DEFAULT buffer cache.

The Buffer Pool Tab of the Memory Detail View calculates the hit ratios for each of the Oracle8 and later buffer caches so you can easily see how often the objects placed into the various caches are being referenced in memory. Examining how often data is satisfied from memory vs. disk will help you determine if the caches are large enough and if they are being used in an optimal manner. The table below describes the information available in this tab

Column	Description
Buffer pool name	The name of the Oracle buffer pool.
Buffer pool hit ratio	The overall hit ratio for the particular cache.

NOTE: The DEFAULT buffer cache will only be shown for those installations not using the specialized caches available in Oracle8 and later.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

The KEEP buffer pool should maintain a hit ratio as close to 100% as possible. However, the buffer pool hit ratio is not near 100% until the database has been up and running in a typical state for a short time.

A poor hit ratio for the RECYCLE buffer pool may not be a bad thing since there is little chance of reusing a block stored in the buffer pool before it is aged out.

NOTE: If you place objects into the KEEP buffer pool, you should periodically reexamine their object sizes to ensure that they are not growing to a physical state that will jeopardize the performance of the KEEP pool.

Library Cache Tab

- [Metrics](#)

The library cache holds commonly used SQL statements - basically database code objects. A method for improving performance in Oracle is to encourage the reuse of SQL statements so expensive parse operations may be avoided. The library cache assists this tuning effort.

The Library Cache tab provides insight into how efficiently the library cache is operating. The table below describes the information available in this section:

Column	Description
Namespace	The region of the library cache.
Gets	The number of times a lock was requested for objects in the particular namespace.
Get Hit Ratio	The percentage of times (with 100% being the maximum) that the object was found in the cache.
Pins	The number of times a pin was requested for objects of this namespace. To see what's happening to Library Cache Pins in a given datasource, you can add this statistic to your Trends view.
Pin Hit Ratio	The percentage of times (with 100% being the maximum) that pin requests were successful.
Reloads	The number of times a piece of the object had to be brought back in from disk to the cache, most likely because it was flushed from the shared pool. To see what's happening to Library Cache Reloads in a given datasource, you can add this statistic to your Trends view.
Invalidations	The number of times objects in this namespace were marked invalid because a dependent object was modified.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

NOTE: An overall high library cache hit ratio is a desirable thing. You should strive for a hit ratio between 95-100%, with 99% being a good performance benchmark for code reuse. When a database is first started, the library cache hit ratio, along with the individual region hit ratios, will not be at an optimal level because all code being used will be relatively new, and as such, must be parsed and placed into the shared pool. If, however, after a solid hour or two of steady database time, the library cache hit ratio has not increased to desirable levels, you should look into the possibility of increasing the `shared_pool_size` parameter.

To keep important code objects from being aged out of the library cache, you can use the `DBMS_SHARED_POOL` package to pin frequently used code objects in memory so they will always be there when needed.

Leading Sessions Tab

- [Metrics](#)

Frequently one or two users cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Sometimes, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system levels.

Performance Center displays information to help you find processes that are using the most memory on the server on the Leading Sessions tab of the Memory Detail view and the Memory tab of the Top Sessions view.

The table below describes the information available on the tabs for all supported database platforms:

Column	Description
User Name	The logon name the session is using.
SID	The unique Oracle identifier for the session.
Serial #	The serial number assigned to the session.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Machine	The name of the client machine name that the session is using.
PGA	The Program Global Area (PGA) is a private memory area devoted to housing the global variables and data structures for a single Oracle process, in KB.
UGA	The User Global Area (UGA) contains session specific information regarding open cursors, state information for packages, database link information, and more. When using Oracle's Multi-threaded Server (MTS), the UGA can be moved up into the SGA, in KB.
Memory Sorts	The number of memory sorts a session has performed.
Total Memory	The amount of memory (PGA + UGA) that the session is consuming, in KB.

NOTE: This information is available on both the Leading Sessions tab of the Memory Detail view and the [Memory tab](#) of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If your database server does not have an overabundance of memory, you should periodically check two things:

- 1 Who are your heavy memory users
- 2 The total percentage of memory each user consumes

If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing.

Objects in Memory Tab

- [Metrics](#)

The Objects in Memory tab of the Memory Detail view shows objects currently in the library cache. This view lets you see exactly which objects are in the shared pool as well as their resource usage and activity levels.

The table below describes the information available on the Objects in Memory tab of the Memory Detail view:

Column	Description
Object Owner	The user account that owns the object.
Object Name	The name of the object.
Type	The type of object: INDEX, TABLE, CLUSTER, VIEW, SET, SYNONYM, SEQUENCE, PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, TRIGGER, CLASS, OBJECT, USER, or DBLINK.
Sharable Memory	The amount of memory consumed by the object in the shared pool.
Loads	The number of times the object has been loaded into the cache. This count increases when an object has been invalidated.
Executions	The number of times that the object has been executed by a session thread.
Locks	The number of users actively locking the object.
Pins	The number of user actively pinning the object.
Kept	Whether or not the object has been pinned in memory with the DBMS_SHARED_POOL package.

NOTE: This information is available on both the Objects in Memory tab of the Memory Detail view and the [Objects in Memory tab](#) of the Objects Detail view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If you see objects with large numbers of loads, this could signal that user sessions frequently use the objects causing the objects to prematurely age out of the cache. To fix this you could increase the size of the shared pool and/or pin the objects with the DBMS_SHARED_POOL package so they do not age out of the library cache.

Sessions Overview Tab

- [Metrics](#)

You can get more detail into dips in the overall buffer cache hit ratio by examining user I/O activity and users' individual cache hit ratios. Frequently you can pinpoint one or more accounts responsible for reducing performance and examine their SQL and other statistics to fix the situation.

The table below describes the information available on the Sessions Overview tab of the Memory Detail view:

Column	Description
Username	The logon name the session is using.
SID	The unique Oracle identifier for the session.
Serial #	The serial number assigned to the session.
Status	The status of the session, ACTIVE or INACTIVE.
Machine	The name of the client machine that the session is using.
O/S ID	The unique operating system identifier for the target session.
Logon Time	The timestamp when the session first logged on.
Blocked	Whether the session is blocked from doing work by another session.
Seconds Idle	The number of seconds since the last time the session actively performed work.
Hit Ratio	The percentage of times the session obtained data from memory vs. physical I/O activity. The maximum value is 100%.

NOTE: This information is available on both the Sessions Overview tab of the Memory Detail view and the [Sessions Overview tab](#) of the Users Detail view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Shared Pool Tab

What are the major components of the shared pool and how much memory does each take up? This question can be answered by viewing the Shared Pool section. You can easily see how much free memory is left in the pool as well as how much space all the other components consume. The table below describes the information available in this section:

Column	Description
Component	The major component of the SGA's shared pool.
Memory	The amount of memory the component is currently using, in bytes.
Percent of Shared Pool	The percentage of the shared pool the component uses.

I/O Statistics

The I/O home page includes the following sections:

- [Archive Files Written Today](#)
- [Active Jobs](#)
- [Active Rollbacks](#)
- [Consistent Reads](#)
- [Index Scans](#)
- [Logical Changes](#)

- [Logical Reads](#)
- [Long Table Scans](#)
- [Physical Reads](#)
- [Physical Writes](#)
- [Redo Log Size](#)
- [Redo Wastage](#)
- [Redo Writes](#)
- [Rollback Gets](#)
- [Rollback Waits](#)
- [Session Leaders - I/O](#)
- [Short Table Scans](#)
- [User Commits](#)
- [User Rollbacks](#)

Related Topics

[I/O Detail](#)

[Home Page Statistics](#)

[Memory Page Statistics](#)

[Objects Page Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

[Users Page Statistics](#)

[Network Statistics](#)

[Contention Statistics](#)

Archive Files Written Today

- [Metrics](#)
- [Troubleshooting](#)

To allow for point-in-time recovery, Oracle writes copies of redo log information to disk. When a database is running in archive log mode, a DBA can (with proper backup techniques in place) recovery nicely from a database error and roll forward to almost any point in time needed, as long as the proper archive logs are in place.

Oracle's ARCH process handles the I/O needed to write these archive logs. The archive files written today statistic provides a count of how many archive files have been written for the current date.

TIP: Click this statistic to drill down to the [Archive view](#).

Metrics

Archive files written more than one per hour or half-hour could indicate a too small redo size (or above average data modification load).

Troubleshooting

If the redo log size is too small for the database, you can redefine to a larger size. You must take care when performing this job and consult the Oracle administrator documentation exact procedures on altering the redo log file size.

User Object Accesses

- [Metrics](#)
- [Troubleshooting](#)

This section of the I/O performance category view details some of the most important statistics relating to object accesses. The table below describes the statistics:

Statistic	Description
Long Table Scans	The number of scans of tables consisting of five or more data blocks. This statistic displays on both the I/O performance category view and the Objects performance category view. For more information on long table scan rows, see Long Table Scan Rows .
Short Table Scans	The number of scans of tables consisting of fewer than five data blocks.
Index Scans	The number of object access by a table's ROWID - usually accomplished through an index. For more information on index scans, see Index (ROWID) Accesses .
User Commits	A user commit, which normally signaling the end of a transaction, causes the LGWR process to write modified blocks to disk.
User Rollbacks	The number of times the user transaction manually issued the times the ROLLBACK SQL statement. Click this statistic to drill down to the Rollback Activity tab of the I/O Detail view.
Active Jobs	The number of active jobs, which are scheduled through the use of the Oracle job queue, indicate how many jobs are currently running in the queue. Click this statistic to drill down to the Active Jobs tab of the I/O Detail view.

Metrics

Depending on the statistics found, here are some general user I/O guidelines:

- Avoid unnecessary scans on large tables. Also, high counts of large table scans are a signal to you as a DBA to both investigate the use of more indexes and to review SQL access through EXPLAIN plans.
- Small table scans are actually a good thing because Oracle can often cache the entire table in a single I/O operation.
- Large numbers of index scans are normally desirable too, because they typically indicate the fastest possible resolution to data access requests.
- Keeping an eye on user commits helps give you one of the truest transaction rate indicators in your database.
- Large numbers of user rollbacks can be undesirable because they indicate that user transactions are not completing for one reason or another.
- An increasing number of active jobs can indicate a backup of job activity in the Oracle job queue.

Troubleshooting

Listed below are some methods you can use to avoid unnecessary large table scans:

- Try not to use SQL statements that include the NOT IN, NOT LIKE, <>, IS NULL operators because they typically suppress the use of indexes.
- When referencing concatenated indexes with queries, be sure that you use the leading column in the index. If you do not, the index will not be used at all.
- Avoid using functions in WHERE predicates. If you must use functions, and you are using Oracle8i, investigate the use of function-based indexes.

If you must use table scans, try and make use of the following Oracle features to increase scan performance as much as possible:

- Parallel query
- Partitioning
- Table CACHE parameter
- KEEP buffer pool (Oracle8)

Active Rollbacks

- [Metrics](#)

Oracle writes data to individual rollback segments to undo changes made to the Oracle database from within a transaction. You can also use these to maintain read consistency for multiple users of modified data. Because Rollback Segments read and write data, they can become very hot areas for I/O. This statistic is a count of how many rollback segments are actively involved in transactions for the database.

NOTE: This statistic displays on both the I/O performance category view and the [Objects performance category view](#).

TIP: Click this statistic to drill down to the [Rollback Activity tab](#) of the I/O Detail view.

Metrics

When you drill down into this statistic and see that most or all rollback segments are involved in transactions, you should consider adding more rollback segments to the system.

Session Leaders - I/O

- [Metrics](#)

Heavy I/O activity in a system sometimes indicates that the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are responsible for 75% or more of the I/O activity. It could be that your system is running a large batch load or other typical processes, which are perfectly okay. On the other hand, it could be a runaway process or other rogue connection, which you need to track down and possibly eliminate.

The leading I/O session's display lets you see who the leading sessions are in your system with respect to I/O.

NOTE: This statistic displays on both the [Users performance category view](#) and the I/O performance category view.

TIP: Click this statistic to drill down to the [Leading Sessions tab](#) of the I/O Detail view.

Metrics

One or two users consuming more than 75% of the total I/O load could indicate a runaway or improper process. You can drill into the I/O activity of all users and quickly see if this is the case.

Redo Wastage and Redo Log Size

- [Metrics](#)
- [Troubleshooting](#)

Oracle redo logs are hotbeds of I/O activity in databases that store heavily modified data. Oracle uses redo log files to perform database recovery in the event of a system crash and it writes redo logs in a cyclical fashion: each log file is filled up before Oracle moves on to the next file. The Redo Log Size statistic reflects the total amount of redo generated in bytes since the last refresh. Redo Wastage indicates the number of bytes that were wasted by redo log buffers being written before they were full.

TIP: Double-click these statistics to drill down to the [LGWR/DBWR Detail tab](#) of the I/O Detail view.

Metrics

A high percentage of wasted bytes to overall redo bytes can help you identify if redo wastage is a problem on your system.

Troubleshooting

Sometimes heavy redo wastage occurs when the `log_checkpoint_interval` parameter is set too high. If you think this is the case for your system, then:

- Edit the `Init.ora` file for the database.
- Increase the amount of `log_checkpoint_interval` to a lower value.
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

Rollback Gets and Rollback Waits

- [Metrics](#)

Oracle writes data to individual rollback segments to undo changes made to the Oracle database from a transaction. You can also use these segments to maintain read consistency of data modified by multiple users. Because rollback segments read and write data, they can become very hot areas for I/O.

A count of rollback gets defines the number of times a process went to a rollback header to glean information. The count of rollback writes is the number of bytes written to all rollback segments.

TIP: Double-click these statistics to drill down to the [Rollback Activity tab](#) of the I/O Detail view.

Metrics

Taking the number of rollback writes and dividing them by the number of rollback gets and waits yields the overall rollback contention ratio, which indicates how bad contention is for rollback segments. A count greater than 1% typically means that you should add more rollback segments to the database.

Physical I/O

- [Metrics](#)
- [Troubleshooting](#)

Physical I/O consists of Oracle going to disk to gather or write data. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. You can also use other processes like the checkpoint and archive processes (CKPT and ARCH). Performance Center shows three key indicators of physical I/O. The table below describes these indicators:

Indicator	Description
Physical Reads	Physical Reads is the total number of physical reads performed on all datafiles since the last refresh.
Physical Writes	Physical Writes is the total number of times the DBWR process has performed writes to various database datafiles since the last refresh
Redo Writes	Oracle's redo logs are hotbeds of I/O activity in databases that store heavily modified data. Redo log files are used to perform database recovery in the event of a system crash. Redo logs are written to in a cyclical fashion - each log file is filled up before Oracle moves on to the next file. The redo writes statistic reflects the total number of redo writes by the LGWR process since the last refresh

NOTE: For more information on physical I/O statistics, see [Logical Vital Signs](#).

TIP: Click this statistic to drill down to the [I/O by Tablespace tab](#) of the I/O Detail view.

Metrics

The table below describes metrics for Physical I/O statistics:

Statistic	Metrics
Physical Reads	Large numbers of physical reads could reflect a too small data/buffer cache. The buffer cache hit ratio is a better indicator of overall logical vs. physical I/O.
Physical Writes	Wait events related to I/O activity are good indicators of physical I/O problems. These events include <code>db file parallel write</code> and <code>db file single write</code> .

Troubleshooting

Doing the following can negate large numbers of continuous physical reads:

- Increasing the size of the data/buffer cache.
- Pinning often-referenced objects in memory by using the KEEP buffer pool (Oracle 8 and higher.)
- Placing heavily scanned objects in larger blocksize tablespaces (16-32KB). For Oracle9i or later.
- Tune SQL statements for better efficiency.

Logical I/O

- [Metrics](#)
- [Troubleshooting](#)

Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. You can also use other processes like the checkpoint and archive processes (CKPT and ARCH). Performance Center shows three key indicators of logical I/O. The table below describes these indicators:

Indicator	Description
Logical Reads	Logical Reads is the total number of db block gets and consistent gets (data read from memory) since the last refresh.
Logical Changes	Logical Changes is the total number of changes that were made to all blocks in the SGA that were part of an update or delete operation. These changes generate redo log entries and are permanent if the transaction is committed. The number of logical changes is an approximate indication of total database work.
Consistent Reads	Consistent Reads is the total number of times a consistent read was requested for a database block. Such a read is performed from Oracle's rollback segments.

NOTE: For more information on logical I/O statistics, see [Logical Vital Signs](#).

TIP: Click this statistic to drill down to the [I/O by Tablespace tab](#) of the I/O Detail view.

Metrics

Regarding raw logical I/O counts, no hard-core metrics exist. However, because physical I/O takes longer to complete than logical (memory) I/O, you should minimize physical read operations when possible. The [buffer cache hit ratio](#) is a better indicator of overall logical vs. physical I/O.

Troubleshooting

While logical I/O is still up to 1,400 times faster than physical disk access, it would be wise to investigate the top logical I/O process using Performance Center and see what SQL it is executing. If one process on the system is consuming between 25-50% of the overall amount, their SQL might require tuning.

I/O Detail

The following tabbed pages are available on the I/O Detail page:

- [I/O by Datafile](#)
- [Log I/O - DBWR/LGWR](#)
- [Rollback Activity](#)
- [I/O by Tablespace](#)
- [Leading Sessions](#)
- [Active Jobs Tab](#)

Active Jobs Tab

- [Metrics](#)

Scheduled using the Oracle job queue, custom batch jobs can be set up to run whenever a DBA would like. The Active Jobs tab of the I/O Detail view presents the active jobs that are running and indicates the number of jobs currently running in the queue.

The table below describes the information available on the Active Jobs tab of the I/O Detail view:

Column	Description
SID	The identifier of the session executing the job.
Job ID	The identifier of the job.
Start Time	The time that the job was started.
Submitted By	The user account that submitted the job.
Run As	The user account whose privileges run the job.
Parse As	The user account that parsed the job.
Next Run	The next scheduled job run time.
Job Contents	The body of the anonymous PL/SQL block that this job executes.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Viewing jobs with long run times could indicate a job caught in a loop or experiencing a problem.

Leading Sessions Tab

- [Metrics](#)

When a system undergoes heavy I/O activity, sometimes you find that all the user connections are contributing somewhat equally to the overall load. Frequently, however, one or two user connections are responsible for 75% or more of the I/O activity. It could be that a large batch load or other typical process is running that is perfectly okay for your system. Alternatively, it could be a runaway process or other rogue connection that you should track down and possibly eliminate.

Performance Center displays information to help you find processes that are using the most memory on the server on the Leading Sessions tab of the I/O Detail view and the I/O tab of the Top Sessions view.

The table below describes the information available on the tabs for all supported database platforms:

Column	Description
Username	The logon name the session is using.
SID	The unique Oracle identifier for the session.
Serial #	The serial number assigned to the session.
Status	The status of the session, ACTIVE or INACTIVE.
Machine	The name of the client machine name that the session is using.
Reads	The number of physical reads.
Writes	The number of physical writes.
Total I/O	The total of all physical I/O operations for the session.

NOTE: This information is available on both the Leading Sessions tab of the I/O Detail view and the [I/O tab](#) of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Pinpointing sessions with abnormally high I/O activity relative to other sessions in the system could aid in ferreting out accounts dragging down overall system performance. You should examine the activity of each session to determine the system workload and to determine if you can reduce the workload or tune the system for better performance.

I/O by Tablespace Tab

- [Metrics](#)

Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database. Other processes like the checkpoint and archive processes (CKPT and ARCH) may also be used.

The Tablespace I/O tab displays details concerning the physical I/O activity at the tablespace level. The table below lists the information available on this tab:

Column	Description
Tablespace	The name of the tablespace.
Physical Reads	The cumulative number of physical reads.
Physical Writes	The cumulative number of physical writes.
Block Reads	The cumulative number of physical block reads.
Blocks Written	The cumulative number of physical block writes.
Read Time	The time spent reading from the tablespace (in hundredths of seconds).
Write Time	The time spent writing to the tablespace (in hundredths of seconds).

Metrics

Generally, you want to see much more logical I/O activity than physical I/O, at least with respect to reads, although this in and of itself is a guarantee of good I/O performance. Seeing logical and physical reads keeping pace with one another is a sure sign that the Oracle SGA's buffer cache is sized too small or that needless, large table scans are occurring, which cause blocks of data to be continually read in and flushed out of the buffer cache.

Other telltale signs of trouble brewing are large amounts of activity visible in user's TEMP tablespaces. The normal interpretation of such a thing is that a large number of disk sorts are taking place (perhaps because the Init.ora/spfile parameter SORT_AREA_SIZE may be set too small).

For more information, see [I/O Home Page](#).

I/O by Datafile Tab

- [Metrics](#)

Physical I/O consists of Oracle going to disk to gather or write data. Logical I/O refers to data access performed in memory. The database writer (DBWR) and log writer (LGWR) processes typically perform all the I/O work in the database against the physical datafile used to hold information. The checkpoint and archive processes (CKPT and ARCH), also perform I/O work in the database.

The Datafile I/O tab displays details concerning the physical I/O activity at the datafile level. The table below lists the information available on this tab:

Column	Description
Datafile	The name of the datafile.

Column	Description
Datafile Status	The availability of the datafile.
Tablespace	The name of the tablespace.
Physical Reads	The cumulative number of physical reads.
Physical Writes	The cumulative number of physical writes.
Block Reads	The cumulative number of physical block reads.
Blocks Written	The cumulative number of physical block writes.
Read Time	The time spent reading from the tablespace (in hundredths of seconds).
Write Time	The time spent writing to the tablespace (in hundredths of seconds).

Metrics

Generally, you will want to see much more logical I/O activity than physical I/O, at least with respect to reads, although this in and of itself is a guarantee of good I/O performance. Seeing logical and physical reads keeping pace with one another is a sure sign that the Oracle SGA's buffer cache is sized too small or that needless, large table scans are occurring, which cause blocks of data to be continually read in and flushed out of the buffer cache.

Other telltale signs of trouble brewing are large amounts of activity visible in user's TEMP tablespaces. The normal interpretation of such a thing is that a large number of disk sorts are taking place (perhaps because the `Init.ora/spfile` parameter `SORT_AREA_SIZE` may be set too small). One last thing to watch is high amounts of activity on datafiles housed on the same directories or file systems. This usually indicates a "hot" drive and could represent disk contention. If you see one or more hot drives, you should consider relocating one or more datafiles to lesser-used physical devices.

For more information, see [I/O Home Page](#).

Rollback Activity Tab

The Rollback I/O tab includes the following sections:

- [Rollback Activity](#)
- [Rollback/Optimal Detail](#)

For more information, see [I/O Home Page](#).

Rollback Activity Grid

- [Metrics](#)

To undo changes made to the Oracle database from within a transaction, Oracle writes data to individual rollback segments. Oracle also uses these segments to maintain read consistency for multiple users of data that is being modified. Because Oracle reads from and writes data to rollback segments, they can become very hot areas for I/O.

The Rollback Activity Detail grid presents everything necessary to view and troubleshoot rollback problems. The first grid displays rollback activity details and the [second grid](#) displays rollback/optimal details. The table below describes the information available in the Rollback Activity Detail grid on the Rollback Activity tab of the I/O Detail view:

Column	Description
Name	The name of the rollback segment.
Size	The size of the rollback segment in KBs.
Shrinks	The number of times the rollback segment has decreased in size.

Column	Description
Extends	The number of times the rollback segment has increased in size.
Gets	The number of header gets (the segment has been used).
Waits	The number of header waits.
Writes	The number of bytes written to the rollback segment.
Active	Indicates whether the rollback segment is active (non zero) or not (zero value).
Status	Indicates the status of the rollback segment, with the two main results being OFFLINE (a segment is offline and unavailable for transactions) and ONLINE (a rollback segment is online and available for transactional use).
High Water Mark	The largest size that the rollback segment has ever grown to.

Rollback/Optimal Detail Grid

- [Metrics](#)

The lower grid presents data on each rollback segment's current status with respect to its optimal size setting (the DBA-imposed size that the segment "should" be).

The table below describes the information available in the Rollback/Optimal Detail grid on the Rollback Activity tab of the I/O Detail view:

Column	Description
Rollback Name	The name of the rollback segment.
Size	The size of the rollback, in bytes.
Optimal Size	The optimal setting for the rollback segment. NULL if the DBA has not imposed an optimal setting on the rollback segment.
High-Water Mark	The largest size to which the rollback segment has ever grown, in bytes.
Pct Above Optimal	The percentage above the optimal setting to which the rollback segment ever grown.

NOTE: If a rollback segment has no OPTIMAL setting, it has NULL values for most values in the Rollback/Optimal Detail grid.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

To properly tune rollback I/O, you must first make sure that you have enough segments to accommodate the workload of the database. Constantly seeing a count of active rollback segments equal to or near the number of rollbacks defined for the database is an indicator that you should create more. An overall rollback contention ratio of 1% or greater is an indicator of too few rollbacks. Seeing wait counts greater than zero for each rollback segment is further evidence that you should create more rollback segments. Oracle9i provides the UNDO tablespace to automatically generate and eliminate the 'correct' number of rollback segments for a system given a certain workload.

After ensuring that enough rollback segments exist in the database, you should then turn your attention to the question of sizing. Dynamic rollback extension can take a toll on performance when rollback segments are consistently enlarged to accommodate heavy transaction loads. Seeing rollback segments undergoing numerous extends and shrinks (as Oracle returns a segment back to its OPTIMAL setting), as well as rollback segments having current or high-water mark sizes greater than their OPTIMAL setting usually is a good indicator that they should be permanently enlarged. Again, Oracle9i's automatic undo management can assist in this process.

DBWR/LGWR Tab

The DBWR/LGWR tab includes Database Writer (DBWR/LGWR) detail and Redo Log detail.

For more information, see [I/O Home Page](#).

Database Writer Detail

- [Metrics](#)

The database writer process (DBWR) handles the flow of information from Oracle's physical datafiles to and from the various memory structures in the system global area (SGA). On platforms that support it, you can configure and use multiple DBWR processes. The log writer (LGWR) process manages the information contained in Oracle's online redo log files and redo log buffer area. On the LGWR/DBWR tab of the I/O Detail view you can view statistics relating specifically to both of these critical Oracle background processes and see if they are running efficiently. The table below lists the information available in this section:

Column	Description
Statistic	The specific metric for the database writer.
Value	The value for the statistic.

Metrics

Of all the statistics presented for the DBWR process, the summed dirty queue length statistic deserves attention. Non-zero values typically indicate buffers left in the write queue after a write request and may indicate that the DBWR process is falling behind. For the LGWR process, nonzero values for the redo log space requests and redo log space wait time statistics could be a cause for concern. Redo log space requests reflect the number of times a user process waited for space in the redo log buffer, while the redo log space wait time is the total wait time in milliseconds. Both could indicate the presence of contention in the redo log buffer. Possible remedies include increasing the LOG_BUFFER size in the SGA.

Redo Wastage

Oracle's redo logs are hotbeds of I/O activity in databases that store heavily modified data. Redo log files are used to perform database recovery in the event of a system crash. Redo logs are written to in a cyclical fashion - each log file is filled up before Oracle moves on to the next file. The Redo Wastage section shows how many bytes were wasted by redo log buffers being written before they were full.

Metrics

Viewing a high percentage of wasted bytes to overall redo bytes can help you identify if redo wastage is a problem on your system.

Steps

Sometimes heavy redo wastage occurs when the log_checkpoint_interval parameter is set too high. If you think this is the case for your system, then

- 1 Edit the Init.ora file for the database.
- 2 Change the amount of log_checkpoint_interval to a lower value.
- 3 Cycle the Oracle server when possible to allow the new value to take effect.
- 4 Monitor new value to see if performance improves.

Space Statistics

The Space performance category view displays the following vital Oracle space statistics:

- [Fragmentation Leaders](#)
- [Schema Leaders - Space](#)
- [Tablespace Overview](#)

Related Topics

[Space Detail](#)

[Home View Statistics](#)

[I/O Page Statistics](#)

[Memory Page Statistics](#)

[Objects Page Statistics](#)

[Users Page Statistics](#)

[Network Statistics](#)

[Contention Statistics](#)

Tablespace Overview

- [Metrics](#)
- [Troubleshooting](#)

The Oracle tablespace is the logical container for the physical storage needs of a database. The tablespace overview displays details for all the tablespaces for a particular database, including their total, used, free space, and percent free, as well as if each can automatically extend to support incoming space requests.

TIP: Click this statistic to drill down to the [Space Usage tab](#) of the Space Detail view.

Metrics

If any tablespace's free space percent amount goes below 10%, and at least one tablespace's datafiles does not have AUTOEXTEND enabled (or the datafile has reached its extend limit), you should take action to ensure that the tablespace does not run out of available free space.

Troubleshooting

There are two things you can do to ensure that a tablespace does not run out of available free space:

- 1 First, you should look into the use of Oracle's AUTOEXTEND feature. AUTOEXTEND lets you give an Oracle tablespace the ability to auto-grow when it has exhausted the free space contained within. You can let a tablespace grow in an unlimited fashion or put constraints on it to stop at a certain point. You can also dictate how much more free space the tablespace gets each time it needs more space than is available. However, AUTOEXTEND enabled for a tablespace does not mean that you cannot run out of space. Remember you still have the physical server limitations to contend with. Make sure you (or your sysadmin) keep a careful eye on the server drives that house your Oracle database files for available free space.
- 2 If the free space on a server drive nears its limit, disable AUTOEXTEND for the datafile(s) that are on the target drive, and use the ALTER TABLESPACE ... ADD DATAFILE command to place a new datafile for the tablespace on another drive that has more free space to offer.

TIP: AUTOEXTEND is not a replacement for proactive space planning. When extra space is needed by the database, and AUTOEXTEND is activated by Oracle, performance slows as Oracle allocates more space for the tablespace. Avoiding AUTOEXTEND aids performance, albeit in a small way.

Fragmentation Leaders

- [Metrics](#)
- [Troubleshooting](#)

The Fragmentation Leaders table shows the tablespaces datafiles suffering from the highest levels of free space fragmentation in the database. Tablespaces are made up of object segments and space extents. Extents are either allocated to object segments or are free. When a tablespace is initially populated, all objects are neatly packed together in the front of the tablespace and all remaining free space is in one free chunk at the end. As objects grow (or extend) they are given new extents of space in the tablespace. If objects are dropped, pockets of free space begin to appear throughout the tablespace. These pockets of space take one of two forms. The table below describes these forms:

Free Space	Description
Honeycombs	Pockets of free space that are adjacent to one another.
Bubbles	Pockets of free space that are trapped between object extents in the tablespace.

TIP: Click this statistic to drill down to the [Fragmentation tab](#) of the Space Detail view.

NOTE: This information is available on both the Fragmentation Leaders table of the Space performance category view and the [Fragmentation tab](#) of the Space Detail view.

Metrics

If you see a tablespace that has many chunks of free space, determine if the tablespace is experiencing honeycomb or bubble fragmentation. You can handle honeycomb fragmentation quite easily, whereas bubble fragmentation is more difficult to solve.

Troubleshooting

You can eliminate honeycomb fragmentation with the ALTER TABLESPACE...COALESCE command. Issuing this command combines all pockets of adjacent free space into one extent. Each database maintenance plan should include a job that coalesces all the free honeycombs in a tablespace into one free chunk. Although Oracle is supposed to perform this operation automatically through the SMON process, it requires you to have the PCTINCREASE parameter of the tablespace set to a nonzero value. Having PCTINCREASE set to a value greater than zero encourages tablespace fragmentation through disparately sized extents. Plus, using SMON in this way is not efficient or entirely reliable.

You can temporarily solve bubble fragmentation by performing a total tablespace reorganization. A better long-term solution for Oracle databases 8.1.5 or later is to convert tablespaces over to locally managed tablespaces. With locally managed tablespaces you either specify the initial extent size and let Oracle automatically size all other extents, or specify a uniform extent size for everything. Problems caused by fragmentation then become a thing of the past.

To help stave off fragmentation problems:

- Set PCTINCREASE to zero for all tablespaces and objects to promote same-sized extents.
- Specify equal-sized allotments for your INITIAL and NEXT object storage parameters.
- Group objects with like growth and storage needs together in their own tablespaces.

You should also avoid fragmentation in the SYSTEM tablespaces. The best ways to do this include:

- Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
- Ensure no user has a quota set for SYSTEM.
- Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

Schema Leaders - Space

- [Troubleshooting](#)

The Space Leaders chart details the schemas in the database that own the most space. In databases where object owners can be many, it is frequently a good idea to take a quick look at which schemas are leading the way in terms of space usage. This is not restricted to production environments only, but can also be extended to dynamic development systems. This is useful when you have many developers in a database that have their own sets of objects. Developers sometimes have a bad habit of creating objects (or copies of objects) that they leave in a database even though they are not using them.

TIP: Click this statistic to drill down to the [Space Usage tab](#) of the Space Detail view.

Metrics

None.

Troubleshooting

If you see a user who owns many objects, or copies of objects, reclaim space in your database by contacting the user to see if those objects can be dropped.

Space Detail

The following tabbed pages are available on the Space Detail page:

- [Fragmentation Tablespace Growth](#)
- [Extents Deficits](#)

- [Object Summary](#)
- [Space Usage](#)
- [Tablespace Map](#)

Tablespace Map Tab

- [Metrics](#)

Tablespaces comprise object segments and space extents. Extents are either free inside a tablespace or allocated to object segments. When a tablespace is initially populated, Oracle neatly packs all objects together in the front of the tablespace and all remaining free space is in one free “chunk” at the end. Unfortunately, this is not how things continue to be in a tablespace. As objects grow, or extend, the database gives them new extents of space in the tablespace. If you drop objects, pockets of free space begin to appear throughout the tablespace. These pockets of space are either honeycombs or bubbles.

The Tablespace Map tab of the Space Detail view lets you see all the extents each object in a tablespace takes up and all the free extents. The map is in block id order, letting you view the tablespace from beginning to end. The table below describes the information available on the Tablespace Map tab of the Space Detail view:

Column	Description
Owner	The owner of the object. NULL if free space occupies the extent.
Object Type	Type of object. NULL if free space occupies the extent.
Object Name	The name of the object occupying the extent. NULL if free space occupies the extent.
Block ID	The starting block number of the extent.
Bytes	The amount of space consumed by the extent, in bytes.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

The tablespace map identifies honeycomb problems with two adjacent chunks of free space. You can eliminate honeycomb fragmentation with the ALTER TABLESPACE...COALESCE command. This command combines all pockets of adjacent free space into one extent. This is important because when an object is newly introduced to a tablespace (or an existing object needs to extend), and a contiguous chunk of free space does not exist to accommodate an object's INITIAL or NEXT size allocation, Oracle must manually coalesce all available honeycombs to try and form a large enough free chunk. This is a performance hit. If possible, you should try to minimize performance hits.

Bubble fragmentation is a more serious matter and is normally only corrected through tablespace or database reorganization. The standard technique is to perform an export of all objects in the tablespace, drop the tablespace, and then import all the objects back into the tablespace. However, this technique may just treat the symptom and not the cause of bubble fragmentation. The real issue is to address the reuse of space extents within a tablespace so that bubble fragmentation does not occur in the first place.

Oracle8 offers the concept of locally managed tablespaces, which can all but eliminate tablespace fragmentation. It totally does away with the storage parameters of MINEXTENTS, MAXEXTENTS, PCTINCREASE, and NEXT. With locally managed tablespaces you either specify the initial extent size and let Oracle automatically size all other extents, or specify a uniform extent size for everything. Problems caused by fragmentation then become outdated.

A few suggestions to manually to help stave off fragmentation problems include:

- Set PCTINCREASE to zero for all tablespaces and objects to promote same-size extents.
- Specify equal-size allotments for your INITIAL and NEXT object storage parameters.

- Group objects with like growth and storage needs together in their own tablespaces.

The SYSTEM tablespace is the major hotbed tablespace for Oracle activities. Therefore, more than any other tablespace, you should avoid fragmentation problems in your SYSTEM tablespace. The easiest way to avoid this is to not allow any user (even the default DBA ID's SYS and SYSTEM) to have access to it. There are three ways to accomplish this:

- 1 Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
- 2 Ensure no user has a quota set for SYSTEM.
- 3 Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

Object Summary Tab

- [Metrics](#)

What takes up space in a database? Simply, objects. In an Oracle database, the primary users of space are tables, indexes, and rollback segments. Under certain circumstances, other objects like materialized views and snapshots can come into play. With respect to space, as a DBA you should keep track of your object's growth and space characteristics. As with tablespaces, you want to keep an eye on fragmentation as well, unless locally managed tablespaces are being used (Oracle 8.1.5 or later).

The Object Summary tab of the Space Detail view displays object sizing information for selected tablespaces. The table below describes the information available on the Object Summary tab of the Space Detail view:

Column	Description
Owner	The user account who owns the object.
Object	The object name.
Object type	The type of object.
Kilobytes	The size of the object, in KB.
Extents	The number of extents used by the object.
Percent of tablespace used	The space in the tablespace consumed by the object, as a percentage.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

To avoid object growth errors in your database, you should consider setting the MAXEXTENTS parameter for your objects (which acts as a growth limit for the object) to a very high value or UNLIMITED if allowed by your version of Oracle. This lets you allocate as many extents as the objects need for growth. It is a good practice to monitor the amount of space your objects use and the number of extents they possess. This clues you in to what the dynamic objects in your database are.

One thing you can face with respect to allowing this type of unchecked extent growth is that your objects can reach a point of needing a new extent of space, but not enough free space exists in its parent tablespace to accommodate the enlargement request. It is good practice to periodically sweep your database for objects that can encounter this error on their next extension.

Understand that allowing unrestricted extent growth can save you from extent allocation errors, but introduces object fragmentation. What can you do if you have objects with high numbers of extents and are not using locally managed tablespaces? For tables and indexes, you have three options:

- 1 Use Oracle's export/import utility to export, drop, and then re-import the fragmented objects back into the database with the export parameter COMPRESS=Y. This brings the object back into the database with one large extent. Take care that large enough free chunks of available free space exist to accept the objects back in, or you can experience allocation errors.
- 2 With Oracle8, you can use the ALTER TABLE...MOVE command to reorganize a table back into one extent in a very safe and efficient manner.
- 3 Use ALTER INDEX...REBUILD to reorganize indexes that have moved into multiple extents.

For rollback segments, you can specify an OPTIMAL setting for each rollback segment. Setting OPTIMAL for the rollback gives the segment the opportunity to shrink back to whatever extent limit you would like to keep the rollback segment. The database periodically does this, but you can manually issue the ALTER ROLLBACK...SHRINK command.

Tablespace Growth Tab

- [Metrics](#)

Growing tablespaces do not have to spell problems for a DBA. Of course, up front planning is the key to sizing tablespaces correctly. Unfortunately, DBAs may not have all the information they need at tablespace creation time (or they may have the wrong information), so a tablespace can approach the end of its free space from time to time. The DBA can allow a tablespace to grow automatically (AUTOEXTEND) to prevent an out-of-space condition. Enabling AUTOEXTEND for a tablespace is quite reassuring for a DBA, but it introduces a new concept for the tablespace: monitoring data file growth. You should monitor your tablespaces that have AUTOEXTEND set so you can get an idea of the growth that is occurring in your database. Monitoring them lets you perform some mini-capacity planning and helps you get a jump-start on server upgrades.

The Tablespace Growth tab displays growth statistics for the database currently being monitored. The table below lists the information available on this tab:

Column	Description
Tablespace Name	The name of the tablespace.
Datafile Name	The name of the tablespace's datafile.
Auto Extend?	Indicates whether the tablespace has one or more datafiles that have the AUTOEXTEND feature set (allowing the datafile to automatically grow).
Original Size	The starting size for the physical datafile.
Current Size	The current size of the physical datafile.
Growth	The percentage of growth experienced for the datafile (if any).

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Using AUTOEXTEND can be especially important for tablespaces devoted to disk sorting. Large data warehouses often must endure large disk sort operations. Having AUTOEXTEND enabled for tablespaces used for temporary segments (sort activity) helps large sort queries to complete when they might otherwise fail due to running out of sort space.

Just having AUTOEXTEND enabled for a tablespace does not mean you cannot run out of space. Remember, you still have to contend with the physical server limitations. Make sure you (or your sysadmin) keep a careful eye on the server drives that house your Oracle database files for available free space. If the free space on a server drive nears its limit, disable AUTOEXTEND for the datafile(s) that are on that drive, and use the traditional ALTER TABLESPACE...ADD DATAFILE command to place a new datafile for the tablespace on another drive that has more free space to offer.

Fragmentation Tab

- [Metrics](#)
- [Troubleshooting](#)

Tablespaces are made up of object segments and space extents. Extents are either allocated to object segments or are free. When a tablespace is initially populated, all objects are neatly packed together in the front of the tablespace and all remaining free space are in one free chunk at the end. Unfortunately, this is not how things continue to be in a tablespace. As objects grow (or extend) they are given new extents of space in the tablespace. If you drop objects, pockets of free space begins to appear throughout the tablespace. These pockets of space are either honeycombs or bubbles. Honeycombs are not so difficult to deal with, but bubbles are another story.

The Fragmentation tab displays two different views of tablespace fragmentation. The first grid displays fragmentation at the tablespace level and the [second grid](#) displays fragmentation at the datafile level. The table below describes the information available in the Tablespace Fragmentation grid:

Column	Description
Tablespace Name	The tablespace name.
Free Space (MB)	The total amount of free space in MB for the tablespace.
Free Chunks	The total number of free chunks in the tablespace.
Largest Chunk (MB)	The largest free chunk (in MB) for the tablespace.

NOTE: Performance Center displays the fragmentation grid in two places, the Space performance category view and the Space Detail view.

The table below describes the information available in the Datafile Fragmentation grid on the Fragmentation tab:

Column	Description
Tablespace	The tablespace name.
Datafile	The name of the datafile.
Datafile Status	The status (AVAILABLE or INVALID) of the individual data file. A status of INVALID means that the file number is not in use. This could happen to a file that was part of a tablespace that was dropped.
Free Space	The amount of free space for the datafile, in MB.
Free Chunks	The number of free chunks in the datafile.
Largest Chunk	The largest free chunk for the datafile, in MB.

Column	Description
Datafile	The name of the datafile.
Autoextend	Indicates whether the datafile can automatically grow in size.
Tablespace	The tablespace name.
Free Chunks	The number of free chunks in the datafile.
Largest Chunk	The largest free chunk (in MB) for the datafile.

NOTE: This information is available on both the [Fragmentation Leaders table](#) of the Space performance category view and the Fragmentation tab of the Space Detail view.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

To spot and correct fragmentation in your tablespaces, you should periodically monitor the fragmentation levels of your tablespaces at a global level. Doing so helps you quickly spot tablespaces that are experiencing fragmentation issues. Seeing a tablespace with only one chunk of free space is a sign that a tablespace is not having fragmentation problems. Seeing a tablespace with a couple of free chunks may not be a big deal either, because the tablespace could be made up of more than one datafile. Each datafile has its own chunk or chunks of free space.

If you see a tablespace that has many chunks of free space, the next thing to do is drill down into it and find out if the tablespace is experiencing honeycomb or bubble fragmentation. Honeycomb fragmentation occurs when pockets of free space exist that are adjacent to one another. Bubbles are pockets of free space that are trapped between object segment extents.

You can eliminate honeycomb fragmentation with the ALTER TABLESPACE...COALESCE command. This command combines all pockets of adjacent free space into one extent. It is important to do this because when an object is newly introduced to a tablespace (or an existing object needs to extend), and a contiguous chunk of free space does not exist to accommodate an object's INITIAL or NEXT size allocation, Oracle must manually coalesce all available honeycombs to try and form a large enough free chunk. This is a performance hit. If possible, you should try to minimize performance hits.

Bubble fragmentation is a more serious matter and is normally only corrected through tablespace or database reorganization. The standard technique is to perform an export of all objects in the tablespace, drop the tablespace, and then import all the objects back into the tablespace. However, this technique may just treat the symptom and not the cause of bubble fragmentation. The real issue is to address the reuse of space extents within a tablespace so that bubble fragmentation does not occur in the first place.

Oracle8 and later offers the concept of locally-managed tablespaces, which can all but eliminate tablespace fragmentation. It totally does away with the storage parameters of MINEXTENTS, MAXEXTENTS, PCTINCREASE, and NEXT. With locally managed tablespaces you either specify the initial extent size and let Oracle automatically size all other extents, or specify a uniform extent size for everything. Problems caused by fragmentation then become a thing of the past.

Troubleshooting

What can you do manually to help stave off fragmentation problems? A few suggestions include:

- Set PCTINCREASE to zero for all tablespaces and objects to promote same-sized extents.
- Specify equal-sized allotments for your INITIAL and NEXT object storage parameters.
- Group objects with like growth and storage needs together in their own tablespaces.

Of all your tablespaces, you want to avoid fragmentation problems in your SYSTEM tablespace the most as this is the major hotbed tablespace for Oracle activities. The easiest way to avoid this is to not allow any user (even the default DBA ID's SYS and SYSTEM) to have access to it. There are three ways to do this:

- 1 Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
- 2 Ensure no user has a quota set for SYSTEM.
- 3 Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

Extents Deficits Tab

- [Metrics](#)

The Extents Deficits tab of the Space Detail view helps you determine if any object in your database is unable to extend into its next extent of space. Sometimes objects have NEXT extent parameters that are too large to fit into the parent tablespace's largest block of free space. For locally managed tablespaces, this is not a problem as the database handles object extent management, but for dictionary-managed tablespaces, the potential for problems exist.

By selecting a tablespace in the Extents Deficits tab, you can find out if any object extent problems are looming on the horizon of your database. The table below describes the information available on the Extents Deficits tab of the Space Detail view:

Column	Description
Owner	The user account that owns the object.
Object Name	The object name.
Object Type	The type of object.
Tablespace Name	The tablespace name.
Next Extent Size	The NEXT extent size of the object in bytes.
Max Contiguous Space	The maximum amount of contiguous space.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

You should ALTER any objects with excessive NEXT extent sizes to reduce the NEXT extent to a manageable size.

If you cannot use locally managed tablespaces to help stave off this problem, manually impose NEXT extent limits for all objects that are well within reason. Additionally, you should set the PCTINCREASE parameter for objects to zero to minimize the potential of stratospheric object growth.

Space Usage Tab

- [Metrics](#)

It is vital that a DBA know how the various accounts assigned to a database consume space. This is true in either production or development databases, because even in development databases, space can become an issue that causes major headaches. If development accounts are creating and cloning many objects without removing them, the result can be a lot of wasted space.

The Space Usage tab of the Space Detail view displays solid information to help with understanding database object ownership. The table below describes the information available on the Space Usage tab of the Space Detail view:

Column	Description
Schema	The name of the user account.
Tablespaces	The number of tablespaces in which the user currently has objects.
Spacial Objects Count	The number of space-related objects that the account owns.
Total Space Used	The amount of space that the account owns, in MB.
Percentage of Database	The database space that this account consumes, as a percentage.

Metrics

Seeing owner account with larger than expected data volumes could warrant further investigation. In addition, seeing accounts that have objects in more tablespaces than they have quota's for should also cause you to examine their account more closely to ensure the correct space privileges are in place.

Objects Page Statistics

The Objects performance category view displays the following vital Oracle objects statistics:

- [Active Rollback Transactions](#)
- [Chained Row Tables](#)
- [Chained Row Fetches](#)
- [Index \(ROWID\) Access](#)
- [Invalid/Unusable Objects](#)
- [Locked Objects](#)
- [Long Table Scan Rows](#)
- [Object Buffer Pool Placement](#)
- [Objects With Space Deficits](#)
- [Objects With Extent Problems](#)
- [Objects Pinned](#)

- [Rollbacks at Optimal](#)
- [Rollbacks Beyond Optimal](#)
- [Total Row Fetches](#)

Related Topics

[Objects Detail](#)

[Home View Statistics](#)

[I/O Page Statistics](#)

[Memory Page Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

[Users Page Statistics](#)

[Contention Statistics](#)

[Network Statistics](#)

Object/Buffer Pool Placement

- [Metrics](#)
- [Troubleshooting](#)

Because data objects can be referenced with different types of usage patterns, Oracle8 offers the option to intelligently place objects into one of three buffer caches. The table below describes the buffer caches:

Buffer Cache	Description
KEEP	Designed to minimize misses in the buffer cache. Small objects that are frequently referenced are candidates for the KEEP buffer pool.
RECYCLE	Designed to avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM, the RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.
DEFAULT	The traditional cache for all other data objects.

NOTE: Unless you specify the KEEP or RECYCLE buffer cache, Oracle automatically places objects into the DEFAULT buffer cache.

TIP: Click this category heading to drill down to the [Buffer Pool tab](#) of the Objects Detail view.

Metrics

When looking at overall database I/O activity, you should keep an eye out for objects that you can place into a particular buffer cache. Consider using a KEEP cache for relatively small, frequently accessed tables that require fast response times. Large tables with random I/O activity and are scanned from beginning to end are good candidates for a RECYCLE cache.

Troubleshooting

Objects can be placed into different buffer pools at object creation time (using the STORAGE... BUFFER_POOL option in Oracle8 or later) or existing objects can be set into a different pool with the ALTER command.

NOTE: Tables, partitions, and indexes can be placed into the different caches.

If you just want to use the DEFAULT buffer pool and not enable any special caches, you can still encourage Oracle to keep certain objects in the cache as long as possible using the CACHE parameter. For example, issuing the command ALTER TABLE...CACHE specifies that the blocks retrieved for a table be placed at the most recently used end of the LRU list in the DEFAULT buffer cache when a full table scan is performed. You can also use the CACHE hint in SQL statements to cache a table, but used in this form, the blocks are only cached until the next time the database is shut down.

Chained Table Placement

- [Metrics](#)
- [Troubleshooting](#)

The chained row fetches statistic represents the rows fetched that were either the chained row type or the migrated row type. The total row fetches statistic represents the total number of rows accessed by user processes.

TIP: Double-click these statistics to drill down to the [Chained Tables tab](#) of the Objects Detail view.

Metrics

If you are seeing a high percentage (30% or greater) of data access operations being satisfied by chained rows, you should take positive steps toward the elimination of chained and migrated rows from your tables.

Troubleshooting

The best way to deal with chained and migrated rows is to prevent them from occurring in the first place. The table below describes some ways to do this:

Method	Solution
Use a Large Block Size	Because chaining and migrations occur when the Oracle block is too small to hold the rows in question, make sure you are using a large enough block size for each database you create. An 8 KB or greater block size is normally recommended to help stave off chained and migrated rows.
Use Proper Values of PCTFREE for Tables	The necessary amount of percent free in a table is what helps prevent row migrations from occurring. If you have a database that houses rows with the potential to grow substantially over their initially inserted size, you can provide a liberal amount of PCTFREE for each of your tables.

If chaining or migrations are already on the rise in your database, you can take steps to counter the attack. Chaining is usually hard to resolve because rows are too big for the originally specified block size. A rebuilt database is normally the only cure for them. Migrations are another story. You can usually back up the rows in your migrated-row tables, re-create them with larger amounts of PCTFREE, and then bring the rows back in. The ANALYZE...LIST CHAINED ROWS command is useful for locating chained and migrated rows in a table.

Active Rollback Transactions

- [Metrics](#)

- [Troubleshooting](#)

To undo changes made to the Oracle database from within a transaction, Oracle writes data to individual rollback segments. You can also use these to maintain read consistency for multiple users of modified data. Because Rollback Segments read and write data, they can become very hot areas for I/O. This statistic is a count of how many rollback segments are actively involved in transactions for the database.

NOTE: This statistic displays on both the [I/O performance category view](#) and the Objects performance category view.

TIP: Click this statistic to drill down to the [Rollback Activity tab](#) of the I/O Detail view.

Metrics

When you drill down into this statistic and see that most or all rollback segments are involved in transactions, you should consider adding more rollback segments to the system.

Troubleshooting

If you are using Oracle8i or earlier, begin by creating new rollback segments and altering them to be online for use. Then monitor the overall active rollback ratio to see if it begins to drop.

For Oracle9i or later, consider the use of UNDO management, which is where Oracle itself automatically manages rollback segments in special UNDO tablespaces.

Index (ROWID) Accesses and Long Table Scan Rows

- [Metrics](#)
- [Troubleshooting](#)

Encouraging index access vs. full table scans on large tables is something that is always sought after. Large tables typically make up more than five data blocks, and are usually accessed better with indexes. Small tables (those with five or fewer data blocks) are surprisingly accessed more efficiently in a full table scan, because Oracle usually caches the entire table in the buffer area of the SGA. The Index (ROWID) statistic counts the number of rows in tables accessed during index access. The Table Scan Rows statistic counts the number of rows accessed for table scan operations.

Metrics

The subject of proper indexing schemes is beyond the scope of this document, but as a DBA you should keep an eye on the counts of index accesses vs. table scans. Mounting counts of large table scans are indicative of a too restrictive indexing policy, and hints that you should revisit the physical design of the database.

Troubleshooting

Here are some ways to avoid unnecessary large table scans:

- Try not to use SQL statements that include the NOT IN, NOT LIKE, < >, IS NULL operators because they typically suppress the use of indexes.
- When referencing concatenated indexes with queries, be sure the leading column in the index is used. If it is not, the index will not be used at all.
- Avoid using functions in WHERE predicates. If functions must be used, and you are using Oracle8i, investigate the use of function-based indexes.

If table scans must be used, try and make use of the following Oracle features to increase scan performance:

- Parallel query

- Partitioning
- Table CACHE parameter
- KEEP buffer pool (Oracle8)

Just adding indexes to a table does not necessarily guarantee increased performance. For normal b-tree indexes to be effective, the selectivity of the column(s) being indexed must be high. Low selectivity columns are good candidates for bitmap indexing.

With either index, make sure that the additional indexes do not overly penalize data modification performance. This can especially be the case with bitmap indexes.

Chained Row Tables

- [Metrics](#)
- [Troubleshooting](#)

In normal circumstances, a row of data should fit completely inside one Oracle block. Sometimes, however, this is not the case and the table suddenly contains chained or migrated rows (rows that span more than one data block).

Chaining occurs when a row is initially too large to fit inside one block. Two or more blocks are used by Oracle to hold the row. Migration deals with rows that have grown so much that they can no longer be contained within their original block. When this occurs, Oracle relocates the row out of its original block into another block, but leaves a pointer behind to indicate the relocation.

Both chaining and migration force Oracle to perform more than one I/O to retrieve data that could normally be obtained with a single I/O operation, with the end result being degraded performance.

TIP: Click this statistic to drill down to the [Chained Tables tab](#) of the Objects Detail view.

Metrics

If the amount of chained rows in your tables exceeds 25-30%, you should take steps to eliminate the problem. Further, if the amount of chained rows accessed in your system vs. total rows accessed approaches 20-30%, you can start the process of eliminating the chained and migrated rows.

Troubleshooting

You can locate tables that contain chained rows. Once found, there are a couple of ways to reorganize tables to remove the chaining/migration problem. However, the best way to deal with chained and migrated rows is to prevent them from occurring. The table below describes two methods:

Method	Description
Use a large block size	Because chaining and migrations occur when the Oracle block is too small to hold the rows in question, make sure you are using a large enough block size for each database you create. An 8KB block size or higher is normally recommended to help stave off chained and migrated rows. If you are using Oracle9i or later, you can create special tablespaces that have larger block sizes (16-32KB) than the overall database block size and place any table that is a candidate for chained/migrated rows into them.
Use proper values of PCTFREE for tables	The necessary amount of percent free in a table helps prevent row migrations from occurring. If you have a database that houses rows with the potential to grow substantially over their initially inserted size, provide a liberal amount of PCTFREE for each of your tables.

If chaining or migrations are already on the rise in your database, you can take steps to counter the attack. Chaining is usually hard to resolve because rows are too big for the originally specified block size. A rebuilt database is normally the only cure for them. Migrations are another story. You can usually back up the rows in your migrated-row tables, re-create them with larger amounts of PCTFREE, and then bring the rows back in. The ANALYZE...LIST CHAINED ROWS command is useful for locating chained and migrated rows in a table.

Objects With Space Deficits

- [Metrics](#)
- [Troubleshooting](#)

If an existing Oracle object needs more space (due to incoming data additions, etc.) Oracle allocates new space based on the object's NEXT extent parameter. If an object's NEXT extent allocation is too large to fit into the parent tablespace's largest contiguous block of free space, and if the underlying tablespace's datafiles do not have their AUTOEXTEND property set or if they have hit their AUTOEXTEND limit, the object cannot grow in size and Oracle issues an error. For locally-managed tablespaces with generous settings for AUTOEXTEND, this problem is nonexistent as object extent management is handled by the database, but this is not the case for dictionary-managed tablespaces.

This statistic displays a count of all objects (including tables, indexes, rollback segments, etc.) that have a defined NEXT extent allocation that is larger than their tablespace's largest contiguous block of free space.

TIP: Click this statistic to drill down to the [Extent Deficits tab](#) of the Space Detail view.

Metrics

You should investigate any count larger than zero, as this indicates a potential object/space problem for the database.

Troubleshooting

There are many solutions available for objects that have problematic NEXT extent sizes:

- You can ALTER the objects to reduce their NEXT extent size to an amount that is smaller than their tablespace's largest contiguous block of free space.
- If the parent tablespace suffers from heavy bubble-style fragmentation, the tablespace can be reorganized so that all pockets of free space are reclaimed.
- The parent tablespace's datafile(s) can have their AUTOEXTEND feature enabled. A side option is to have the datafile growth AUTOEXTEND option set to UNLIMITED.
- The object can be moved to a locally-managed tablespace or the parent tablespace can be converted to locally-managed (Oracle8i or later). If locally-managed tablespaces cannot be used, you can manually impose NEXT extent limits for all objects that are well within reason. In addition, you should set the PCTINCREASE parameter for objects to zero to minimize the potential of stratospheric object growth. To better control all object size/parameter allocations, you should set growth parameters for all objects at the tablespace level and not the object level.

Objects With Extent Problems

- [Metrics](#)
- [Troubleshooting](#)

Objects With Extent Problems helps you determine how many objects in your database are at or approaching maximum extents. For locally managed tablespaces, this is not a problem as the database handles object extent management, but for dictionary-managed tablespaces, the potential for problems exist.

Metrics

Use the Object Extents tab of the Object Detail view to investigate which objects are at or near max extents.

Troubleshooting

If you cannot use locally managed tablespaces to help stave off this problem, increase the number of max extents for the object.

Objects Pinned

- [Metrics](#)
- [Troubleshooting](#)

The Library Cache is an area of memory in Oracle's shared pool that permits users to share and reuse code objects like SQL statements, PL/SQL procedures, functions, packages, and the like. To avoid expensive parse operations, the Library Cache was introduced to let Oracle cache the code objects and shared SQL so other user processes can share the same SQL or PL/SQL.

TIP: Click this statistic to drill down to the [Shared Pool tab](#) of the Objects Detail view.

Metrics

Objects that have many loads into the shared pool but are not marked KEPT are ideal candidates for being permanently pinned in memory using Oracle's DBMS_SHARED_POOL utility. Doing so helps reduce expensive load operations.

Troubleshooting

Use the DBMS_SHARED_POOL utility to pin an Oracle code object in the shared pool. Keep in mind that pinning objects in memory with the DBMS_SHARED_POOL utility is only good for the duration of the current Oracle instance's life. Once you bring Oracle down, and then start back up, you need to re-pin the objects again. You can add this operation to your database maintenance plan, which is invoked on any database startup.

Invalid/Unusable Objects

- [Metrics](#)
- [Troubleshooting](#)

The statistic combines the number of objects with a status of INVALID with the number of indexes that have a status of UNUSABLE. Objects like procedures, packages, functions, triggers, and views can become invalidated for a variety of reasons. The main cause is generally a dependent object that has been altered or removed from the system. However, other objects, like indexes, can become invalid also due to scenarios like SQL*Loader problems. If an object that has become invalid is still referenced (through an application or SQL query tool), a variety of problems can result. Sometimes Oracle reports a clear error stating the problem, while other times seemingly odd behavior is exhibited by the database.

TIP: Click this statistic to drill down to the [Invalid Objects tab](#) of the Objects Detail view.

Metrics

There is no reason to have invalid objects in a production database. If your production databases have invalid objects that are no longer needed, promptly remove them from each system. Any needed objects that are indicating an invalid status should quickly be fixed before access problems develop.

It is very normal for development databases to have invalid objects because developers create, modify, and compile objects all the time. The only invalid object that really should not be present in either a development or production database is an invalid index.

Troubleshooting

If code objects have become invalidated, you can issue an ALTER ... COMPILE command to see if they compile properly and become valid once again. If they do not, then check the USER_ERRORS view for any resulting error messages. Indexes can be validated once more by using the ALTER INDEX ... REBUILD command.

Locked Objects

- [Metrics](#)

Locked Objects is a count of all objects on the system that currently have some form of lock against them.

TIP: Click this statistic to drill down to the [All Locks tab](#) of the Locks view.

Metrics

None.

NOTE: Drilling down into the count of locked objects displays detail on each object that is locked, along with the user process holding the lock and the type of lock held.

Rollback Summary

- [Metrics](#)
- [Troubleshooting](#)

These statistics display how many rollback segments are currently setting at their OPTIMAL size setting and how many have extended into a larger size. Dynamic rollback extension can take a toll on performance when rollback segments are consistently enlarged to accommodate heavy transaction loads.

TIP: Double-click these statistics to drill down to the [Rollback Activity tab](#) of the I/O Detail view.

Metrics

Seeing rollback segments undergoing numerous extends and shrinks (as Oracle returns a segment to its OPTIMAL setting), as well as rollback segments having current or high watermark sizes greater than their OPTIMAL setting usually is a good indicator that they should be permanently enlarged.

Troubleshooting

For rollback segments, you can specify an OPTIMAL setting for each rollback segment. Setting OPTIMAL for the rollback gives the segment the opportunity to be shrunk back to the extent boundary where you would like to keep the rollback segment. This is periodically done by the database, but can also be accomplished by manually issuing the ALTER ROLLBACK...SHRINK command.

Objects Detail

The following tabbed pages are available on the Objects Detail page:

- [Invalid Objects](#)

- [Buffer Pool](#)
- [Chained Tables](#)
- [High Watermarks](#)
- [Objects Accessed](#)
- [Object Extents](#)
- [Objects in Memory](#)

Invalid Objects Tab

- [Metrics](#)

Objects like procedures, packages, functions, triggers, and views can become invalidated for a variety of reasons, with the main cause being a dependent object that has been altered or removed from the system. However, other objects, like indexes, can become invalid also due to scenarios like SQL*Loader problems. If an object that has become invalid is still referenced (through an application or SQL query tool), a variety of problems can result. Sometimes Oracle will report a clear error stating the problem, while other times seemingly quirky behavior will be exhibited by the database. In any event, as a DBA you should be on the lookout for objects in your database that have suddenly become invalid.

The Invalid Objects tab of the Objects Detail view displays invalid objects found in the database. The [first grid](#) displays a summary of the invalid/unusable objects. The [second grid](#) displays the details of the invalid/unusable objects. The table below describes the information available on the Invalid/Unusable Objects Summary grid on the Invalid Objects tab of the Objects Detail view:

Column	Description
Owner	The user account that currently owns invalid database objects.
Count	The number of invalid objects found in the user account.

The table below describes the information available on the Invalid/Unusable Objects Detail grid on the Invalid Objects tab of the Objects Detail view:

Column	Description
Owner	The user account that owns the objects.
Object Name	The name of the invalid object.
Object Type	The type of object: PROCEDURE, VIEW, etc.
Status	The status of the object, INVALID or UNUSABLE.
Created	The date and time stamp when the object was created.
Last DDL Date	The last structural modification date for the object.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

There is no reason to have invalid objects in a production database. If your production databases have invalid/unusable objects that are no longer needed, you should promptly remove them from each system. Any needed objects that are indicating an invalid status should quickly be fixed before access problems develop.

Correcting invalid objects like procedures and views often involves performing an ALTER...COMPILE operation. If the object status does not return to VALID, then further examination is warranted.

It is very normal for development databases to have invalid objects because developers will no doubt be creating, modifying, and compiling objects all the time. The only invalid object that really should not be present in either a development or production database is an invalid index.

Buffer Pool Tab

- [Metrics](#)

Because data objects can be referenced with different types of usage patterns, Oracle8 offers the option to intelligently place objects into one of three distinct buffer caches. The table below describes these buffer caches:

Buffer Cache	Description
KEEP	Minimizes misses in the buffer cache. Small, frequently referenced objects are candidates for the KEEP buffer pool.
RECYCLE	Lets you avoid having large numbers of infrequently accessed blocks of data crowd out objects that need to be referenced in RAM. The RECYCLE pool is good for large objects that are scanned from beginning to end without the need for keeping all their data in RAM.
DEFAULT	The traditional cache for all other data objects.

NOTE: Unless you specify the KEEP or RECYCLE buffer cache, Oracle automatically places objects into the DEFAULT buffer cache.

The [first grid](#) displays a summary of the objects in the buffer pool. The [second grid](#) displays the details of the objects in the buffer pool. The table below describes the information available on the Buffer Pool Objects Summary grid on the Buffer Pool tab of the Objects Detail view:

Column	Description
Object Type	The type of object (TABLE, INDEX, etc.)
Buffer Pool	The name of the buffer pool (KEEP, RECYCLE, DEFAULT).
Object Count	The number of objects in the indicated pool for the object type.

The table below describes the information available on the Buffer Pool Objects Detail grid on the Buffer Pool tab of the Objects Detail view:

Column	Description
Buffer Pool	The name of the buffer pool (KEEP, RECYCLE, DEFAULT).
Owner	The account that owns the named object.
Object Name	The name of the object.
Object Type	The type of object (TABLE, INDEX, etc.)

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

When looking at overall database I/O activity, you should keep an eye out for objects that you can place into a particular buffer cache. Consider using a KEEP cache for relatively small, frequently accessed tables that require fast response times. Large tables with random I/O activity and are scanned from beginning to end are good candidates for a RECYCLE cache.

Objects can be placed into different buffer pools at object creation time (using the STORAGE...BUFFER_POOL option) or objects can be moved into a different pool at any time with the ALTER command.

If you just want to use the DEFAULT buffer pool and not enable any special caches, you can still encourage Oracle to keep certain objects in the cache as long as possible using the CACHE parameter. For example, issuing the command ALTER TABLE...CACHE specifies that the blocks retrieved for a table be placed at the most recently used end of the LRU list in the DEFAULT buffer cache when a full table scan is performed. You can also use CACHE hint in SQL statements to cache a table, but used in this form, the blocks will only be cached until the next time the database is shut down.

Chained Tables Tab

- [Metrics](#)

In normal circumstances, a row of data should fit completely inside one Oracle block. Sometimes, however, this is not the case and the table suddenly finds itself containing chained or migrated rows, which are rows that span more than one data block.

Chaining occurs when a row is initially too large to fit inside one block. Two or more blocks are used by Oracle to hold the row. Migration deals with rows that have grown so much that they can no longer be contained within their original block. When this occurs, Oracle relocates the row out of its original block into another block, but leaves a pointer behind to indicate the relocation. Both chaining and migration force Oracle to perform more than one I/O to retrieve data that could normally be obtained with a single I/O operation, resulting in degraded performance.

The Chained Tables tab of the Objects Detail view displays tables that currently have chained rows. Note that for truly accurate data to be returned, the ANALYZE command needs to be routinely inside the database. This ensures that the data dictionary is up to date with respect to object demographics.

The table below describes the information available on the Chained Tables tab of the Objects Detail view:

Column	Description
Schema	The user account that owns the chained tables.
Table Name	The table name.
Rows	The number of rows that reside inside the table.
Chained Rows	The number of chained rows that reside inside the table.
Percent Chained	The percentage of chained rows to total rows in the table.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

How bad should chaining or migration get before you take action with a table? If the amount of chained rows in your tables exceeds 25-30%, you should take steps to begin eliminating the problem. Further, if the amount of chained rows accessed in your system vs. total rows accessed approaches 20-30%, you can start the process of eliminating the chained and migrated rows.

The best way to deal with chained and migrated rows is to prevent them from occurring in the first place. The table below describes some ways to do this:

Method	Solution
Use a Large Block Size	Because chaining and migrations occur when the Oracle block is too small to hold the rows in question, make sure you are using a large enough block size for each database you create. An 8 KB or greater block size is normally recommended to help stave off chained and migrated rows.

Method	Solution
Use Proper Values of PCTFREE for Tables	The necessary amount of percent free in a table is what helps prevent row migrations from occurring. If you have a database that houses rows with the potential to grow substantially over their initially inserted size, you can provide a liberal amount of PCTFREE for each of your tables.

If chaining or migrations are already on the rise in your database, you can take steps to counter the attack. Chaining is usually hard to resolve because rows are too big for the originally specified block size. A rebuilt database is normally the only cure for them. Migrations are another story. You can usually back up the rows in your migrated-row tables, re-create them with larger amounts of PCTFREE, and then bring the rows back in.

High Watermarks Tab

- [Metrics](#)

Tables that are the victim of much insert-delete activity can become performance problems due to their block high watermarks. A table's high watermark equals the last block in the table that was used to hold data. The problem is that the high watermark is not reset by DELETE activity, so it is possible for a table to have absolutely no data in it, but contain a high watermark that is many blocks high. When such a table is scanned, Oracle reads up to the high watermark even if no rows exist in the table at all. This can make for some unnecessarily large scan times.

The table below describes the information available on the High Watermarks tab of the Objects Detail view:

Column	Description
Owner	The user account that owns the tables.
Table Name	The name of the table.
Size (bytes)	The physical size of the table, in bytes.
Rows	The number of rows occupying the table.
Blocks	The number of blocks the table consumes.
Empty Blocks	The number of blocks assigned to the table that contain no data.
High Watermark	The last block in the table that was used to hold data.

NOTE: For truly accurate data to be returned, the ANALYZE command needs to be routinely run on the viewed tables. This ensures that the data dictionary is up to date with respect to object demographics.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Red flags should go up to you if you observe a table that contains no rows (or few rows) but has a high watermark that is nonzero for no rows, and much above zero for a few. Again, to effectively detect a table's high watermark, you must make timely use of the ANALYZE command to update the data dictionary. Doing so ensures a fresh supply of statistics available to analyze used and free data blocks in a table.

Objects Accessed Tab

- [Metrics](#)

There are two things to consider regarding the performance of objects and Oracle memory use. First, because data can be accessed many times faster in RAM than from disk, you should keep often-referenced data from your objects in memory as much as possible. It is also wise to keep object definitions in memory so that references to them are quick.

Second, to avoid expensive parsing operations and continuous loads into the Oracle shared pool, you should do everything possible to encourage code reuse and ensure that parsed code is ready and available in the library cache.

The Objects Accessed tab of the Objects Detail view displays what objects are currently being accessed in memory and users who are using them. The table below describes the information available on the Objects Accessed tab of the Objects Detail view:

Column	Description
Owner	The user account that owns the object.
Object name	The name of the object in memory.
Type	The type of object (TABLE, CURSOR, VIEW, etc.)
Accessing User	The user who is currently using the object.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Once you have an idea of what objects your users are most often after, you can do a couple of things to help their access process along.

With respect to data objects, you can use the Oracle8 concept of the KEEP buffer cache to force Oracle to hold often referenced data. The KEEP buffer cache is ideal for holding small lookup tables and the like. If you have an earlier version of Oracle or don't want to divide your current buffer cache, you can alternatively use the CACHE table attribute to encourage Oracle to keep data blocks of cached tables at the most recently used end of the LRU buffer cache chain.

Objects that have many loads but are not marked KEPT in the shared pool are ideal candidates for being permanently pinned in memory using the DBMS_SHARED_POOL utility. Doing so helps reduce expensive load operations. Keep in mind that pinning objects in memory with the DBMS_SHARED_POOL utility is only good for the duration of the current Oracle instance's life. Once Oracle is brought down and started back up, re-pin the objects.

Object Extents Tab

- [Metrics](#)

Object fragmentation results when objects consume multiple extents of space in a tablespace rather than a single block of space. Although performance problems with respect to object fragmentation are not as severe as they were in older versions of Oracle, response-time penalties can still be chalked up to this situation. When multiple extents exist for an object, the amount of time it takes Oracle to scan it can be longer than if the object was made up of only one extent. This typically holds true when extents are scattered on different parts of the physical server disk. In addition, a performance hit is taken each time an object must extend into another extent of space.

The Object Extents tab of the Objects Detail view displays objects whose extent count has exceeded a user-suggested numerical limit. If objects are found, Performance Center displays them on the this tab. The table below describes the information available on the Object Extents tab of the Objects Detail view:

Column	Description
Owner	The user account that owns the object.
Object Name	The name of the object.
Object Type	The type of object (TABLE, INDEX, etc.)
Extents	The number of extents for the object.
Limit	The MAXEXTENTS limit imposed on the object by the DBA.
Pct/Limit	How close the object is to reaching its extent limit, in percentage.

NOTE: For more information see, [Object Extents Tab](#).

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

When is object fragmentation a problem for an object? Opinions on this vary widely. As a general rule of thumb, if an object is exhibiting response time degradations, and no other factor can be attributed to the slowdown, examine the object to see how badly fragmented it is. If extent numbers exceed 20, you should consider reorganizing the object back into a single extent of space. This rule of thumb does not apply to objects assigned to locally managed tablespaces as extent fragmentation is expected and not a performance issue.

The best preventive for this problem is specifying the correct allocation of space for the object in the first place, but what can you do if you have objects with high numbers of extents? For tables and indexes you have three options:

- 1 Use Oracle's export/import utility to export, drop, and import the fragmented objects back into the database with the export parameter COMPRESS=Y. This brings the objects back into the database with one large extent. Make sure that large enough chunks of available free space exist to accept the object back in, or you may experience space allocation errors.
- 2 With Oracle8, you can use the ALTER TABLE...MOVE command to reorganize a table back into one extent in a very safe and efficient manner.
- 3 Use ALTER INDEX...REBUILD to reorganize indexes that have moved into multiple extents.

For rollback segments, you can specify an OPTIMAL setting for each rollback segment. Setting OPTIMAL for the rollback gives the segment the opportunity to be shrunk back to the extent boundary where you would like to keep the rollback segment. This is periodically done by the database, but can also be accomplished by manually issuing the ALTER ROLLBACK...SHRINK command.

Objects in Memory Tab

- [Metrics](#)

The Objects in Memory tab of the Objects Detail view shows objects currently in the library cache. This view lets you see exactly which objects are in the shared pool as well as their resource usage and activity levels.

The table below describes the information available on the Objects in Memory tab of the Objects Detail view:

Column	Description
Owner	The user account that owns the object.
Object Name	The name of the object.
Type	The type of object: INDEX, TABLE, CLUSTER, VIEW, SET, SYNONYM, SEQUENCE, PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, TRIGGER, CLASS, OBJECT, USER, or DBLINK.
Sharable Memory	The amount of memory consumed by the object in the shared pool.
Loads	The number of times the object has been loaded into the cache. This count increases when an object has been invalidated.
Executions	The number of times that the object has been executed by a session thread.
Locks	The number of users actively locking the object.
Pins	The number of users actively pinning the object.
Kept	Whether or not the object has been pinned in memory with the DBMS_SHARED_POOL package.

NOTE: This information is available on both the [Objects in Memory tab](#) of the Memory Detail view and the Objects in Memory tab of the Objects Detail view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If you see objects with large numbers of loads, this could signal that the objects are being frequently by user sessions and can age out of the cache prematurely. To fix this you could increase the size of the shared pool and/or pin the objects with the DBMS_SHARED_POOL package so they do not age out of the library cache.

OS Page Statistics

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. The OS Statistics page of Performance Center lets you examine operating system metrics for the following platforms:

- AIX
- HP-UX

NOTE: To view processor info and swap disk info on an HP-UX box, you need to login as ROOT in the OS login.

- Linux
- Solaris
- Unix
- Windows XP, 2000, and NT

CAUTION: The OS statistics are not available from the Web Client.

- [Average Disk Queue](#)
- [Network Output Queue \(Windows\)](#)
- [Network Queue \(Solaris\)](#)
- [Page Faults/Sec](#)
- [Processor Queue](#)
- [Processor Speed](#)
- [Processor](#)
- [Available Paged Memory \(Windows\)](#)
- [Available Physical Memory](#)
- [Available Swap Memory \(AIX, HP-UX, Linux, Solaris, Unix\)](#)
- [Total Paged Memory \(Windows\)](#)
- [Total Physical Memory](#)
- [Total Swap Memory \(AIX, HP-UX, Linux, Solaris, Unix\)](#)
- [Free Disk Space](#)
- [Total Disk Space](#)
- [Used Disk Space](#)
- [Number of Logins](#)
- [Number of Processes](#)
- [Number of Processors](#)
- [Top CPU Process](#)
- [Top I/O Process](#)
- [Top Memory Process](#)

Processor Time

The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

Metrics

If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.

Troubleshooting

Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

Processor Speed

The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Processor

The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Disk Time

The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

Metrics

You should avoid consistently seeing values for this statistic greater than 90%.

Troubleshooting

Add more disk drives and partition the files among all of the drives.

Load Average

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Metrics

High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Paged Memory Used

The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

NOTE: This statistic is available for the Windows platform.

Metrics

This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.

Troubleshooting

Increase the size of page file.

Number of Processors

This statistic displays the number of processors currently in use.

Swap Memory Used

The Swap Memory Used statistic is the percentage of swap space currently in use.

Metrics

If the percentage of swap memory used is above 90%, you may be running out of memory.

Troubleshooting

Increase the size of your swap files.

Average Disk Queue

The Average Disk Queue statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Metrics

This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.

Troubleshooting

Some things you can do if you have problems with this statistic include:

- Redistribute the data on the disk with the large average disk queue to other disks.
- Upgrade to faster disk(s).

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics

This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue

The Processor Queue Length statistic is the number of threads in the processor queue.

Metrics

Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting

A sustained high value in the Processor Queue could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Network Output Queue/Network Queue

The Network Output Queue Length statistic is the number of threads in the processor queue.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics

Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting

A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Available Physical Memory

The Available Physical Memory statistic represents the amount of RAM available to all processes.

Metrics

This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Troubleshooting

Some things you can do if you have problems with this statistic include:

- Check the running processes to see if there are any memory leaks.
- Stop any services that are not required.
- Install additional RAM.

Available Paged Memory

The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

NOTE: This statistic is available for the Windows platform.

Metrics

If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.

Troubleshooting

Increase the size of page file.

Available Swap Memory

The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Metrics

If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.

Troubleshooting

Increase the size of swap files.

Total Physical Memory

The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Metrics

This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Total Paged Memory/Total Swap Memory

The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics

It is recommended that this value is between 1.5 to 3 times the amount of RAM on the system.

Used Disk Space

The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Total Disk Space

Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two:

- 1 If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- 2 You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Free Disk Space

The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Top Memory Process

Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics

If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Processes Overview

The Processes Overview of the OS Summary includes the following sections:

- [Top CPU Process](#)
- [Top I/O Process](#)
- [Top Memory Process](#)

Top CPU Process

Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics

If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.

Troubleshooting

Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process

The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Top Memory Process

The top I/O process identifies the Oracle process that currently is using the highest percentage of memory in the database.

Metrics

To obtain more details on the top memory process, locate the SID in the Top Sessions grid and drill down to obtain more granular information.

Number of Logins

This statistic displays the total number of logins on the server.

Number of Processes

This statistic displays the total number of processes on the server.

CPU Tab

The CPU tab of the OS Detail includes the following sections:

- [Context Switches/Sec](#)
- [CPU Utilization](#)
- [Interrupts/Sec](#)
- [Processor Queue Length](#)

CPU Utilization

The CPU Utilization section includes the following information:

- [% Privileged Time](#)
- [% User Time](#)

% Privileged Time

The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

NOTE: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

Metrics

The ideal range should be 0-40% (less than 40% indicates excessive system activity).

Troubleshooting

If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

% User Time

The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Metrics

If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

Interrupts/Sec

The Interrupts/Sec statistic is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. It does not include deferred procedure calls (DPCs), which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every ten milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Metrics

The ideal range should be 0-5000. A number greater than 5000 indicates possible excessive hardware interrupts; justification is dependent on device activity.

Context Switches/Sec

The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Metrics

The ideal range should be between 0-10,000. GA number greater than 10,000 may indicate too many threads contending for resources.

Processor Queue Length

The Processor Queue Length statistic is the number of threads in the processor queue. There is a single queue for processor time even on computers with multiple processors.

NOTE: For Windows systems, unlike the disk counters, this counter shows ready threads only, not threads that are running.

Metrics

A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Processes Tab

The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

Column	Description
Process	The name of the process.
User	The user of the process.
ID	The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
CPU	The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.
User Mode	The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.
Memory WINDOWS ONLY	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory (MB)	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory	Memory is the percentage of the memory used of the total memory.
Active Memory	Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.

Column	Description
I/O Data	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
Elapsed Time	The total elapsed time, in seconds, that this process has been running.
Thread Count	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Handle Count	The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.
Priority	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Creating Proc ID	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Page Faults/Sec	Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.
Page File	Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private	Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.

I/O Tab

The table below describes the information available in this section:

Column	Description
Disk	The disk number assignment.
Reading (KB/s)	The amount of bytes read from the device.
Writing (KB/s)	The amount of bytes written to the device.
Disk Read Time	Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Disk Write Time	Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Disk Time	Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.
Avg. Read Queue	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
Avg. Write Queue	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Disk Reads/Sec	Disk Reads/Sec is the rate of read operations on the disk.
Disk Writes/Sec	Disk Writes/Sec is the rate of write operations on the disk.

Memory Tab

The Memory tab of the OS Detail page includes the following sections:

- [Cache Efficiency](#)
- [Free Physical](#)
- [Free Paged](#)
- [Paging Activity](#)
- [Page Faults](#)
- [Total Physical](#)
- [Total Paged](#)

Paging Activity

The Paging Activity section includes the following statistics:

- [Pages Input/Sec](#)
- [Pages Output/Sec](#)

Pages Input/Sec

The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Metrics

This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Pages Output/Sec

The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Metrics

Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Free Physical

The Free Physical statistic is the amount of physical memory that is uncommitted.

Metrics

None.

Free Paged

The Free Paged statistic is the amount of uncommitted virtual memory.

Metrics

None.

Total Physical

The Total Physical statistic is the total physical memory available.

Metrics

None.

Total Paged

The Total Paged statistic is the amount of total virtual memory, in bytes. Used Memory is the physical memory that has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.

Metrics

None.

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics

This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Cache Efficiency

The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

- [Copy Read Hits%](#)
- [Data Map Hits%](#)
- [MDL Read Hits%](#)
- [Pin Read Hits%](#)

Copy Read Hits %

The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

Metrics

When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Data Map Hits %

The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %

The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %

The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab

The Space tab of the OS Detail page includes the following sections:

- [Disk Space Free](#)
- [Disk Space Detail](#)

Disk Space Free

The Disk Space Free metric displays the amount of free disk space in megabytes.

Metric

None.

Disk Space Detail

The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Local Filesystem	The name of the file system.
Type	The type of file system.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.

Statistic	Description
Capacity	The percentage of space used on the device.
Mounted On	The mount point of the device.

Network Tab

The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

Column	Description
Network Interface	The name of network interface.
INET Address/Address	The IP address assigned to the network interface.
Pkts Sent/Sec	The number of packets sent per second.
Pkts Received/Sec	The number of packets received per second.
Sent (KB/Sec)	The number of bytes sent per second.
Received (KB/Sec)	The number of bytes received per second.
Out Pkts Discarded	The number of outbound packets discarded.
In Pkts Discarded	The number of inbound packets discarded.
Out Pkt Errors	The number of outbound packet errors.
In Pkt Errors	The number of inbound packet errors.
Queue Length	The queue length.
Collisions	The number of collisions.
Packets Discarded	The number of packets discarded.

Contention Statistics - Oracle

The Contention performance category view displays the following vital Oracle contention statistics:

- [Buffer Busy Wait Ratio](#)
- [Enqueue Waits](#)
- [Free List Waits](#)
- [Latch Immediate Miss Ratio](#)
- [Latch Miss Ratio](#)
- [Latch Sleep Ratio](#)
- [Max Query Slaves](#)
- [Parallel Query Busy Ratio](#)

- [Prolonged Lock Waits](#)
- [Redo Log Space Requests](#)
- [Redo Log Wait Time](#)
- [Rollback Contention Ratio](#)
- [Sessions in Buffer Busy Waits](#)
- [Sessions Waiting for Latches](#)
- [Used Query Slave](#)
- [Users Blocked](#)
- [Users Waiting \(General\)](#)

Latch Miss Ratio

- [Metrics](#)
- [Troubleshooting](#)

The latch miss ratio defines the number of times a process obtained a willing-to-wait latch vs. missing the attempt. Latches protect the many memory structures in Oracle's SGA. They ensure that one and only one process at a time runs or modifies any memory structure at the same instant. Latches are much more restrictive than locks, which at least allow for some collective user interaction. They have no queuing mechanism, and therefore, you either get the latch or you are forced to continually retry.

TIP: Click this statistic to drill down to the [System Waits tab](#) of the Contention Detail view.

Metrics

If the latch miss ratio exceeds 1%, you should take action to resolve the amount of latch contention occurring.

Troubleshooting

You should examine the details regarding the latch contention. Increasing the `shared_pool_size` can also assist in resolving latch problems. Here are a few specialized cases of latch contention that you can easily rectify:

Latch Contention Case	Description	Resolution
Cache buffer chain latch	This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch.	You can alleviate contention somewhat by adding more buffers to the cache (through the <code>db_block_buffers</code> parameter) or by adding more LRU latch chain latches with the <code>db_block_lru_latches</code> parameter.
Library cache latches	Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.	If possible, make SQL more reusable by using bind variables instead of literals. Doing so should alleviate contention for the library cache latch."

Other routine latch contention problems used to include the redo allocation and redo copy latches, however in Oracle 8.1.5 or later these are generally obsolete.

Latch Immediate Miss Ratio

- [Metrics](#)
- [Troubleshooting](#)

The latch immediate miss ratio defines the number of times a process obtained a not-willing-to-wait latch vs. missing the attempt. Latches protect the many memory structures in Oracle's SGA. They ensure that one and only one process at a time runs or modifies any memory structure at the same instant. Latches are much more restrictive than locks, which at least allow some collective user interaction. They have no queuing mechanism, and therefore, you either get the latch or you are forced to continually retry.

TIP: Click this statistic to drill down to the [Latch Detail tab](#) of the Contention Detail view.

Metrics

If the latch immediate miss ratio exceeds 1%, you should take action to resolve the amount of latch contention occurring.

Troubleshooting

You should examine the details regarding the latch contention. Increasing the `shared_pool_size` can also assist in resolving latch problems. Here are a few specialized cases of latch contention that you can easily rectify:

Latch Contention Case	Description	Resolution
Cache buffer chain latch	This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch.	You can alleviate contention somewhat by adding more buffers to the cache (through the <code>db_block_buffers</code> parameter) or by adding more LRU latch chain latches with the <code>db_block_lru_latches</code> parameter.
Library cache latches	Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.	If possible, make SQL more reusable by using bind variables instead of literals. Doing so should alleviate contention for the library cache latch."

Other routine latch contention problems previously included the redo allocation and redo copy latches. In Oracle 8.1.5 or later these are generally obsolete.

Latch Sleep Ratio

- [Metrics](#)
- [Troubleshooting](#)

Latches protect the many memory structures in Oracle's SGA by ensuring that one and only one process at a time runs or modifies any memory structure at the same instant. Latches are much more restrictive than locks, which at least allow for some collective user interaction. They have no queuing mechanism, and therefore, you either get the latch or you are forced to continually retry.

A sleep indicates that a latch could not be obtained for a process, and that Oracle continues to retry. A high ratio indicates that many processes had to sleep multiple times before obtaining a requested latch.

TIP: Click this statistic to drill down to the [Latch Detail tab](#) of the Contention Detail view.

Metrics

You should keep sleeps as low as possible. If the overall sleep ratio exceeds 1%, you should take action to resolve the amount of latch contention that is occurring.

Troubleshooting

You should examine the details regarding the latch contention. Increasing the `shared_pool_size` can also assist in resolving latch problems. Here are a few specialized cases of latch contention that you can easily rectify:

Latch Contention Case	Description	Resolution
Cache buffer chain latch	This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch.	You can alleviate contention somewhat by adding more buffers to the cache (through the <code>db_block_buffers</code> parameter) or by adding more LRU latch chain latches with the <code>db_block_lru_latches</code> parameter.
Library cache latches	Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of literals being used for SQL statements instead of bind variables.	If possible, make SQL more reusable by using bind variables instead of literals. Doing so should alleviate contention for the library cache latch."

Other routine latch contention problems previously included the redo allocation and redo copy latches. In Oracle 8.1.5 or later these are generally obsolete.

Buffer Busy Wait Ratio

- [Metrics](#)
- [Troubleshooting](#)

TIP: Click this statistic to drill down to the [Buffer Busy Waits tab](#) of the Contention Detail view.

Metrics

Buffer busy waits normally center around contention for rollback segments, too small an `INITRANS` setting for tables, or insufficient free lists for tables.

Troubleshooting

On Oracle8i or earlier, the remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for `INITRANS` to allow for more transactions per data block, and more free lists.

For Oracle9i or later, you can use the automatic segment management feature in Oracle9i locally-managed tablespaces to help make free list problems a thing of the past. Using an `UNDO` tablespace in 9i or later can help remedy any rollback contention problem.

You can also obtain which objects have actually experienced buffer busy waits in Oracle9i or later by querying the `sys.v_$segment_statistics`. This view is not populated unless the configuration parameter `statistics_level` is set to `TYPICAL` or `ALL`.

Rollback Contention Ratio

- [Metrics](#)

- [Troubleshooting](#)

Rollback segments are used by the Oracle RDBMS to hold data needed to rollback (or undo) any changes made through inserts, updates, or deletes to various Oracle objects. They also allow Oracle to have read consistency for long running queries. Rollback segments are used for recovery purposes and they play a role during exports of database information.

In a heavy transaction processing environment, rollback segments are accessed continuously and therefore are subject to contention problems. The rollback contention ratio helps identify contention occurring on the system relating to rollbacks.

TIP: Click this statistic to drill down to the [Rollback Waits tab](#) of the Contention Detail view.

Metrics

Overall, if the rollback segment wait ratio approaches 1% or more, you should consider creating more rollback segments. You should also think about creating a specialized, larger, rollback segment to be used by long running transactions. Doing so alleviates dynamic rollback extensions and cuts down heavily on ORA-01555 snapshot too old errors.

Troubleshooting

Begin by creating new rollback segments and altering them to be online for use. Then monitor the overall contention ratio to see if it begins to drop.

Users Blocked

- [Metrics](#)
- [Troubleshooting](#)

On a small system, a single blocking user has the potential to stop work for nearly all other processes. On larger systems, this can cause major headaches. Although Oracle supports unlimited row-level locking, blocking lock situations do occur. User processes holding exclusive locks and not releasing them via a proper COMMIT frequency, generally cause most blocks.

TIP: Click this statistic to drill down to the [Blocking Locks tab](#) of the Locks view.

Metrics

You should investigate any blocking lock statistic indicator above zero to prevent a mushrooming situation.

Troubleshooting

You can quickly remedy a blocking lock situation by issuing a KILL against the offending process. This eliminates the user's stranglehold on the accessed objects and lets other user processes complete. Tools like Performance Center make it easier to discover the blocked lock situation, the tricky part is preventing the blocking lock situation in the first place.

The culprit of blocking lock scenarios is usually the application design, or the SQL used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT...FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible, something not always easy to do.

Users Waiting (General)

- [Metrics](#)
- [Troubleshooting](#)

User connections that are waiting on a system generally occur for two reasons:

- A process waits because a requested resource is not available.
- A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) should not worry you. The two wait causes mentioned above are worth your time and investigation.

NOTE: This statistic displays on both the Contention performance category view and the [Users performance category view](#).

TIP: Click this statistic to drill down to the [Session Waits tab](#) of the Contention Detail view.

Metrics

To determine the actual wait causes user connections are experiencing, drill down from the global count of users waiting into the actual system and user wait details of your target database. This lets you locate the exact causes of currently experienced waits.

Troubleshooting

If you find a problem, drill down into wait details to determine whether the waits are resource related.

Sessions Waiting for Latches

- [Metrics](#)

User connections that are waiting on a system generally occur for two reasons:

- 1 A process waits because a requested resource is not available.
- 2 A process waits for Oracle to perform a prerequisite task for its given operation.

Sessions Waiting for Latches displays the number of user processes specifically waiting for latches.

TIP: Click this statistic to drill down to the [Latch Waits tab](#) of the Contention Detail view.

Metrics

If you find nonzero values for this statistic, you should access the drill-down views to discover the exact latches that are making users wait.

Sessions in Buffer Busy Waits

- [Metrics](#)

User connections that are waiting on a system generally occur for two reasons:

- A process waits because a requested resource is not available.
- A process waits for Oracle to perform a prerequisite task for its given operation.

Sessions involved in buffer busy waits indicates contention for data blocks in the buffer cache.

TIP: Click this statistic to drill down to the [Buffer Busy Waits tab](#) of the Contention Detail view.

Metrics

If you find nonzero values for this statistic, you should access the drill-down views to discover the exact latches that are making users wait.

Prolonged Lock Waits

- [Metrics](#)
- [Troubleshooting](#)

This statistic shows the number of sessions that have been blocked for over 300 seconds.

On a small system, a single blocking user has the potential to stop work for nearly all other processes. On larger systems, this can cause major headaches. Although Oracle supports unlimited row-level locking, blocking lock situations do occur. User processes holding exclusive locks and not releasing them via a proper COMMIT frequency generally cause most blocks. Click this statistic to drill down to the [Blocking Locks tab](#) of the Locks view.

Metrics

You should investigate any indicator above zero to prevent a mushrooming situation.

Troubleshooting

You can quickly remedy a blocking lock situation by issuing a KILL against the offending process. This eliminates the user's stranglehold on the accessed objects and lets other user processes complete. Performance Center make it easier to discover the blocked lock situation, but the tricky part is preventing the blocking lock situation in the first place.

The culprit of blocking lock scenarios is usually the application design, or the SQL used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible, something not always easy to do.

Redo Log Space Requests

- [Metrics](#)
- [Troubleshooting](#)

The Oracle RDBMS is able to manage recovery by recording all changes made to a database through the use of redo log files. Oracle writes modifications made to a database to the redo log files, which you can archive to another medium for disaster recovery. The background process that performs these operations is Oracle's Log Writer (LGWR). There is a buffer area in Oracle's System Global Area (SGA) that is used to reduce redo log file I/O, whose size, or lack thereof, can affect performance in a busy system. Sometimes a user process must wait for space in this redo log buffer. Oracle uses the log buffer area to cache redo entries prior to writing them to disk, and if the buffer is not large enough for the redo entry load, waits can occur.

TIP: Click this statistic to drill down to the [LGWR/DBWR Detail tab](#) of the I/O Detail view.

Metrics

The two main numbers to watch are:

- 1 Redo log space requests
- 2 Redo log wait time

If either statistic strays too far from zero, increase the `log_buffer` parameter and add more memory to the redo log buffer.

Troubleshooting

If you find a problem, do the following:

- 1 Edit the `Init.ora` file for the database.
- 2 Increase the amount of `log_buffer` to a higher value (take care not to over-allocate; ensure enough free memory exists on the server before increasing the value).
- 3 Cycle the Oracle server when possible to allow the new value to take effect.
- 4 Monitor the new value to see if performance improves.

When adjusting the `log_buffer` parameter, make sure that you make the amount a multiple of the block size. Otherwise, on database startup, Oracle returns an error stating that you have entered an invalid amount for the redo log buffer.

NOTE: Oracle silently increases the `log_buffer` parameter if you make it smaller than its default size for a given platform.

Redo Log Space Wait Time

- [Metrics](#)
- [Troubleshooting](#)

The Oracle RDBMS is able to manage recovery by recording all changes made to a database through the use of redo log files. Oracle writes modifications made to a database to the redo log files, which you can archive to another medium for disaster recovery. The background process that performs these operations is Oracle's Log Writer (LGWR). There is a buffer area in Oracle's System Global Area (SGA) that is used to reduce redo log file I/O, whose size, or lack thereof, can affect performance in a busy system. Sometimes a user process must wait for space in the redo log buffer. Oracle uses the log buffer to cache redo entries prior to writing them to disk. If the buffer area is not large enough for the redo entry load, waits can occur. While waits can be an important indicator of contention, a better measure is the amount of wait time that your users experience.

TIP: Click this statistic to drill down to the [LGWR/DBWR Detail tab](#) of the I/O Detail view.

Metrics

The two main numbers to watch are:

- 1 Redo log space requests
- 2 Redo log wait time

If either statistic strays too far from zero, increase the `log_buffer` parameter and add more memory to the redo log buffer.

Troubleshooting

If you find a problem, do the following:

- 1 Edit the Init.ora file for the database.
- 2 Increase the amount of log_buffer to a higher value (take care not to over-allocate; ensure enough free memory exists on the server before increasing the value).
- 3 Cycle the Oracle server when possible to allow the new value to take effect.
- 4 Monitor the new value to see if performance improves.

When adjusting the log_buffer parameter, make sure you make the amount a multiple of the block size. Otherwise, on database startup, Oracle returns an error stating that you have entered an invalid amount for the redo log buffer.

NOTE: If you make the log_buffer parameter smaller than its default size for a given platform, Oracle silently increases it.

Parallel Query Busy Ratio

- [Metrics](#)
- [Troubleshooting](#)

Oracle's parallel query feature, when used properly, allows large increases in performance for many SQL and utility operations. Parallel operations can be introduced through SQL hints, specified in an object's DDL, or used in command line arguments for utility programs like SQL*Loader. To effectively service parallel query requests, you must ensure that enough query servers exist in the database instance. The parallel query busy ratio is an indicator of how busy all the servers are on the database in question.

TIP: Click this statistic to drill down to the [Parallel Query tab](#) of the Contention Detail view.

Metrics

If the parallel query busy ratio approaches 80-90%, you should think about:

- Adding more query servers to the database.
- Examining parallel requests to ensure they are being used in an efficient and necessary manner.

Troubleshooting

If you find a problem, do the following:

- 1 Edit the Init.ora file for the database.
- 2 Increase the amount of parallel_max_servers to a higher value.
- 3 Cycle the Oracle server when possible to allow the new value to take effect.
- 4 Monitor the new value to see if performance improves.

You can also investigate the use of the parallel_automatic_tuning parameter in Oracle 8.1 or later.

Used Query Slaves

- [Metrics](#)
- [Troubleshooting](#)

Oracle's parallel query feature, when used properly, allows large increases in performance for many SQL and utility operations. Parallel operations can be introduced through SQL hints, specified in an object's DDL, or used in command line arguments for utility programs like SQL*Loader. To effectively service parallel query requests, you must ensure that enough query servers exist in the database instance. The used query slave statistic displays how many query servers are actively performing work in a database.

TIP: Click this statistic to drill down to the [Parallel Query tab](#) of the Contention Detail view.

Metrics

If the parallel query busy ratio approaches 80-90%, or you see all the available slaves being used, you should think about:

- Adding more query servers to the database.
- Examining parallel requests to ensure they are being used in an efficient and necessary manner.

Troubleshooting

If you find a problem, do the following:

- 1 Edit the Init.ora file for the database.
- 2 Increase the amount of parallel_max_servers to a higher value.
- 3 Cycle the Oracle server when possible to allow the new value to take effect.
- 4 Monitor new value to see if performance improves.

You can also investigate the use of the parallel_automatic_tuning parameter in Oracle 8.1 or later.

Max Query Slaves

- [Metrics](#)
- [Troubleshooting](#)

Oracle's parallel query feature, when used properly, allows large increases in performance for many SQL and utility operations. Parallel operations can be introduced through SQL hints, specified in an object's DDL, or used in command line arguments for utility programs like SQL*Loader. To effectively service parallel query requests, you must ensure that enough query servers exist in the database instance. The max query slave statistic displays the maximum number of query slaves allowed on the system as dictated by the parallel_max_servers parameter.

TIP: Click this statistic to drill down to the [Parallel Query tab](#) of the Contention Detail view.

Metrics

If the parallel query busy ratio approaches 80-90%, or you see all the available slaves being used, you should think about:

- Adding more query servers to the database.
- Examining parallel requests to ensure they are being used in an efficient and necessary manner.

Troubleshooting

If you find a problem, do the following:

- 1 Edit the Init.ora file for the database.
- 2 Increase the amount of parallel_max_servers to a higher value.

- 3 Cycle the Oracle server when possible to allow the new value to take effect.
- 4 Monitor the new value to see if performance improves.

Free List Waits

- [Metrics](#)

Free lists are lists of Oracle data blocks that contain free space for an Oracle object. Every table has at least one free list. You can use Free lists to locate free blocks of space when a request is made of a table for the insertion of a row. Free list contention can reduce the performance of applications when many processes are involved in the insertion of data to the same table.

TIP: Click this statistic to drill down to the [System Waits tab](#) of the Contention Detail view.

Metrics

If free list contention approaches 1%, you should add additional free lists to the most dynamic database objects through the use of the STORAGE parameter.

Enqueue Waits

- [Metrics](#)
- [Troubleshooting](#)

An enqueue is an advanced locking device that lets multiple database processes share certain resources. Enqueue waits typically occur when sessions wait to be granted a requested lock. Sometimes these locks are internal Oracle locks while other times they could be locks for rows of data in a table.

NOTE: Enqueues are issued implicitly by Oracle.

TIP: Click this statistic to drill down to the [System Waits tab](#) of the Contention Detail view.

Metrics

You should investigate any enqueue waits that read consistently above one or more (delta statistics).

Troubleshooting

Removing contention for enqueues is usually an application design issue. Many enqueue waits are either contention for certain rows in the database or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that your SQL does not tarry over rows in the database during modification operations. If it does, you should tune your SQL.

Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).

Contention Detail View

The following tabbed pages are available on the Contention Detail view:

- [Buffer Busy Waits](#)
- [Latch Detail](#)
- [Latch Waits](#)
- [Parallel Query](#)
- [Rollback Waits](#)
- [Session Waits](#)
- [System Waits](#)

Buffer Busy Waits Tab

- [Metrics](#)

Buffer busy waits occur when a process needs to access a data block in the buffer cache, but cannot, because it is being used by another process. Therefore, it must wait. The Buffer Busy Waits tab of the Contention Detail view shows how long users are waiting for buffers in the buffer cache. The table below describes the information available on the Buffer Busy Waits tab of the Contention Detail view:

Column	Description
Username	The user account being used by the session.
SID	The unique Oracle identifier for the session.
OSUser	The operating system ID of the user session.
Machine	The client machine name used by the session.
Program	The program being executed against the Oracle database by the session.
Total Waits	The total number of buffer busy waits for the session.
Time Waited	The total amount of time waited for the event, in hundredths of a second.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.

Latch Detail Tab

- [Metrics](#)

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry.

The Latch Detail tab of the Contention Detail view presents a detailed view of latch activity. The table below describes the information available on the Latch Detail tab of the Contention Detail view:

Column	Description
Name	The name of the latch.
Gets	The number of times the latch was requested by a process.
Misses	The number of failed attempts to acquire the latch on the first attempt.
Spins	The number of times a failed first attempt acquired the latch on a spin.
Immediate Gets	The total number of nowait requests for a latch.
Immediate Misses	The total number of failed nowait attempts to acquire the latch on the first attempt.
Sleeps	The total number of requests that "paused" while waiting for a latch.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Common indicators of latch contention are a latch miss ratio (which records willing-to-wait mode latch requests) and latch immediate miss ratio (which records no-wait mode latch requests). These statistics reflect the overall health of how often latch requests were made and satisfied without waiting. If either of these exceeds 1%, latch contention can be causing system slowdowns.

The table below describes two latch contention situations that you can recognize and quickly fix:

Situation	Remedy
Cache Buffer Chain Latch	This latch is responsible for protecting paths to database block buffers in the buffer cache. Very high I/O loads tend to cause contention for this latch. You can alleviate contention somewhat by adding more buffers to the cache (through the <code>db_block_buffers</code> parameter) or by adding more LRU latch chain latches with the <code>db_block_lru_latches</code> parameter.
Library Cache Latches	Protects cached SQL statements in the library cache area of the Oracle shared pool. Contention for this latch is the usual result of using literals for SQL statements instead of using bind variables.

NOTE: Other routine latch contention problems included the redo allocation and redo copy latches, but these have pretty much been made obsolete in Oracle 8.1.5 or later.

Latch Waits Tab

- [Metrics](#)

Protecting the many memory structures in Oracle's SGA are latches. They ensure that one and only one process at a time can run or modify any memory structure at the same instant. Much more restrictive than locks (which at least allow for some collective user interaction), latches have no queuing mechanism so either you get it or you do not and are forced to continually retry. The Latch Waits tab of the Contention Detail view shows detailed information about current latch wait situations. The table below describes the information available on the Latch Waits tab of the Contention Detail view:

Column	Description
Username	The user account being used by the session.
SID	The unique Oracle identifier for the session.
Latch Name	The name of the latch.
Parameter1 Text	The description of the first wait parameter.
Parameter1	The first wait/tuning parameter.
Parameter2 Text	The description of the second wait parameter.
Parameter2	The second wait/tuning parameter.
Seq #	The sequence number that uniquely identifies the wait. It is incremented for each wait.
Wait	The time waited, in hundredths of a second.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Common indicators of latch contention are a latch miss ratio (which records willing-to-wait mode latch requests) and latch immediate miss ratio (which records no-wait mode latch requests). These statistics reflect the overall health of how often latch requests were made and satisfied without waiting. By using latch information on this view (contained in the parameter1 and parameter2 boxes) you can locate the exact object experiencing contention.

Parallel Query Tab

- [Metrics](#)

Oracle's parallel query option offers great potential in pursuit of faster response times. The ability to split a request into multiple segments that are processed in parallel can greatly speed up the race to a result, however it is important that Oracle be set up with the proper resources necessary to use parallel functions in an efficient manner. The Parallel Query tab of the Contention Detail view displays information relating to parallel query performance. The table below describes the information available on the Parallel Query tab of the Contention Detail view:

Column	Description
Parallel Query Statistic	The name of the statistic affecting parallel query performance.
Value	The statistical value.

Metrics

Seeing a server's busy statistic that nears or matches the maximum values allotted for parallel query slaves is a sure sign that more query slaves need to be added to the system. Also, seeing a server's highwater value that matches the value for the PARALLEL_MAX_SERVERS parameter can indicate a need to raise the number of query slaves given to Oracle.

Rollback Waits Tab

- [Metrics](#)

In systems with heavy data modification loads, the rollback segments can become quite a hot spot. Contention for rollback segments can occur between competing database transactions and cause a performance slowdown.

The Rollback Waits tab of the Contention Detail view displays all the rollbacks available on the system and their wait statistics. The table below describes the information available on the Rollback Waits tab of the Contention Detail view:

Column	Description
Name	The name of the rollback segment.
Gets	The number of header gets (the segment has been used).
Waits	The number of header waits.
WaitRatio	The individual contention ratio for the rollback segment.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Overall, if the global rollback contention ratio approaches 1% or more, you should consider creating more rollback segments. Seeing many individual rollback segments with wait ratios above 1% could also be a cause for concern.

You should also think about creating a specialized, larger, rollback segment for use by long running transactions. Doing so alleviates dynamic rollback extensions and cuts down dramatically on ORA-01555 snapshot-too-old errors. For a database under heavy DML loads, consider creating two rollback tablespaces on separate drives to minimize disk contention among the rollback segments.

Session Waits Tab

- [Metrics](#)

Session contention is merely a subset of contention that is viewable at the global database level. Often, it takes analysis at the session level to pinpoint the exact source of contention that is occurring globally. Therefore, you need to become accustomed to viewing contention statistics and waits at the user process level.

When monitoring waits with respect to user sessions, there are two areas of interest:

- 1 What HAS the user session been waiting on?
- 2 What IS the user session waiting on?

Oracle records both sets of wait statistics for you. In reviewing previous waits for a session, you can see what types of things have caused a bottleneck in the session. The table below describes the information available on the Sessions Waits tab of the Contention Detail view:

Column	Description
User	The user account being used by the session.
SID	The unique Oracle identifier for the session.
Wait Cause	The name of the wait cause
Program	The program being executed against Oracle by the user session.
Time (sec)	The number of seconds that process has been waiting.
Wait State	The value of the wait state. WAITING indicates that the session is waiting. WAITED UNKNOWN TIME indicates that the duration of last wait is unknown. WAITED SHORT TIME indicates that the last wait was less than 1/100th of a second. WAITED KNOWN TIME indicates that the wait is equal to the time of the last wait.
Object Owner	The name of the object owner.
Object Name	The name of the object.

NOTE: This information is available on both the [Session Waits tab](#) of the Users Detail view and the Session Waits tab of the Contention Detail view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

To view the wait times for sessions and overall system wait events, you must set the TIMED_STATISTICS parameter to TRUE for your Oracle databases. You can either set this parameter in your Init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Some waits can be ignored like SQL*Net more data from client and SQL*Net message from client. Others like enqueue waits can indicate a lock contention problem. Waits, like db file scattered reads, can indicate sessions involved in long table scan operations.

System Waits Tab

- [Metrics](#)

Waits on a system generally occur for three reasons:

- 1 A process waits because it has no work to do.
- 2 A process waits because a requested resource is not available.
- 3 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) should not worry you at all, however the other two wait causes are the ones worth your time and investigation. From a global database level, there are many different types of waits and sources of contention. The System Waits tab of the Contention Detail view presents all the various system waits that have occurred on the system since startup. The table below describes the information available on the System Waits tab of the Contention Detail view:

Column	Description
Wait Event	The name of the wait event.
Total Waits	The total number of waits for the event.
Total Timeouts	The total number of timeouts for the event.
Time Waited	The total amount of time waited for the event in hundredths of a second.
Average Wait Time	The average amount of time waited for the event in hundredths of a second.
Percent Total	The total percentage of waits this event makes up relative to all waits on the system.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Which waits should concern you and which waits can you ignore (for the most part)? Common wait events to ignore include:

- SQL*Net more data from client
- SQL*Net message from client
- client message
- rdbms ipc message
- pipe get
- pmon timer
- smon timer
- Null event

If you see many enqueue waits, this normally indicates either contention for certain rows in the database, or the result of database-initiated lock escalation. You should examine the use of indexes to make sure all referencing foreign keys are indexes and that your SQL is tuned to not tarry over rows in the database during modification operations. Enqueue waits can also be the result of space management tasks (such as objects extending) and disk sorts (mainly in tablespaces that do not make use of the TEMPORARY tablespace parameter).

Buffer busy waits normally center around contention for rollback segments, too small an INITRANS setting for tables, or insufficient free lists for tables. The remedy for each situation would be increasing the number of rollback segments, or altering tables to have larger settings for INITRANS to allow for more transactions per data block, and more free lists.

An interesting wait event is the db file scattered read event. This event is indicative of table scans occurring on the system. Large numbers of them can indicate heavy scan activity and the need to revisit your indexing/physical design.

Network Statistics - Oracle

The Network performance category view displays the following vital network statistics:

- [Bytes Received from Client](#)

- [Bytes Received from DBLink](#)
- [Bytes Sent to Client](#)
- [Bytes Sent to DBLink](#)
- [MTS Request Activity](#)
- [MTS Response Activity](#)
- [Network Contention](#)
- [Roundtrips to/from Client](#)
- [Roundtrips to/from DBLink](#)

Bytes Sent to Client

- [Metrics](#)

The total number of bytes sent over the network to all Oracle client machines from the database since the last refresh.

Metrics

All users connected to a database via a client/server connection generate network traffic, so seeing decent volumes of network traffic is normal. Out of the ordinary numbers should raise alarms because possible network saturation could occur.

Bytes Received from Client

- [Metrics](#)

The total number of bytes sent over the network to the Oracle database from all client machines since the last refresh.

Metrics

All users connected to a database via a client/server connection generate network traffic, so seeing decent volumes of network traffic is normal. Out of the ordinary numbers should raise alarms because possible network saturation could occur.

Roundtrips to/from Client

- [Metrics](#)

The total number of Oracle network messages sent to and received from the client.

Metrics

All users connected to a database via a client/server connection generates network traffic, so seeing decent volumes of network traffic is normal. Out of the ordinary numbers should raise alarms because possible network saturation could occur.

Bytes Sent to DBLink

- [Metrics](#)

The total number of bytes sent by all client machines over the network to any database link since the last refresh.

Metrics

All users connected to a database via a client/server connection generate network traffic, so seeing decent volumes of network traffic is normal. Out of the ordinary numbers should raise alarms because possible network saturation could occur.

NOTE: Database link traffic could experience longer response times depending on the distributed nature and set up of the involved databases.

Bytes Received from DBLink

- [Metrics](#)

The total number of bytes received by all client machines over the network from any database link since the last refresh.

Metrics

All users connected to a database via a client/server connection generate network traffic, so seeing decent volumes of network traffic is normal. Out of the ordinary numbers should raise alarms because possible network saturation could occur.

NOTE: Database link traffic could experience longer response times depending on the distributed nature and setup of the involved databases.

Roundtrips to/from DBLink

- [Metrics](#)

The number of Oracle network messages sent over and received from a database link.

Metrics

All users connected to a database via a client/server connection generate network traffic, so seeing decent volumes of network traffic is normal. Out of the ordinary numbers should raise alarms because possible network saturation could occur. Also note that database link traffic could experience longer response times depending on the distributed nature and setup of the involved databases.

Network Contention

- [Metrics](#)
- [Troubleshooting](#)

Sometimes after sending a request to the database, a user experiences a slowdown in getting a request back from the Oracle server. These delays are usually negligible with respect to the actual amount of time that the user process actually waits. Examining the amount of network-related waits in light of all waits experienced by the database indicates the percentage of waits that the network is contributing to the overall contention scenario.

Metrics

The main thing to do with respect to your database and the network is to minimize the amount of data that travels over it. This means ensuring that only the amount of data your users truly need is being returned to their query tools or front-end application.

Troubleshooting

Eliminating unnecessary large volumes of rows being sent over the wire to clients from the Oracle database is one place to start, as overall traffic can be reduced.

MTS Response Activity

- [Metrics](#)
- [Troubleshooting](#)

In the past, database professionals avoided Oracle's Multi-threaded server option unless they had a server that was low on horsepower. Instead, most people used the direct option of each user having his or her own SQL*Net thread. This was (and probably still is) the fastest way for Oracle networking to perform its job.

The MTS Response Activity area displays information regarding the amount of waits in the MTS response queues managed by Oracle's shared dispatcher processes.

NOTE: These statistics are only available for Oracle servers with the Multi-threaded server option.

Metrics

When the number of waits for MTS responses begins to climb, adding more shared dispatchers can increase performance.

Troubleshooting

You can add more dispatchers to Oracle both by editing the Init.ora file and changing the `mts_dispatchers` and `mts_max_dispatchers` parameters to higher amounts (and then cycling Oracle) or dynamically changing the `mts_dispatchers` parameter through use of the ALTER SYSTEM command.

MTS Request Activity

- [Metrics](#)
- [Troubleshooting](#)

In the past, database professionals avoided Oracle's Multi-threaded server option unless they had a server that was low on horsepower. Instead, most people used the direct option of each user having his or her own SQL*Net thread. This was (and probably still is) the fastest way for Oracle networking to perform its job.

The MTS Request Activity area displays information regarding the amount of waits in the MTS request queues managed by Oracle's shared dispatcher processes.

NOTE: These statistics are only available for Oracle servers with the Multi-threaded server option.

Metrics

When the number of waits for MTS requests begins to climb, performance can be increased by adding more shared dispatchers.

Troubleshooting

You can add more dispatchers to Oracle by both editing the Init.ora file and changing the mts_dispatchers and mts_max_dispatchers parameters to higher amounts (and then cycling Oracle) or dynamically changing the mts_dispatchers parameter through use of the ALTER SYSTEM command.

Users Page Statistics

The Users home page includes the following sections:

- [Active Connections](#)
- [Current Locks](#)
- [Current Transactions](#)
- [Inactive Connections](#)
- [Leading Sessions - CPU](#)
- [Leading Sessions - I/O](#)
- [Leading Sessions - Memory](#)
- [Max Connections](#)
- [Max Open Locks](#)
- [Max Sessions](#)
- [Max Transactions](#)
- [Open Cursors](#)
- [Sessions Involved in Disk Sorts](#)
- [Total Sessions](#)
- [Users Blocked](#)
- [Users Waiting](#)

Related Topics

[Users Detail](#)

[Home View Statistics](#)

[I/O Page Statistics](#)

[Memory Page Statistics](#)

[Objects Page Statistics](#)

[OS Page Statistics](#)

[Space Page Statistics](#)

[Top SQL Page Statistics](#)

Active Connections

The Active Connections statistic is the total number of active and open threads currently reported in the database as well as the number of processes actively performing work.

Metrics

None.

TIP: Click this statistic to drill down to the [Sessions Overview tab](#) of the Users Detail view.

Current Locks and Max Open Locks

- [Metrics](#)
- [Troubleshooting](#)

This set of statistics reflects the total number of DML locks currently held on the database as well as the maximum number of DML locks allowed on the system at one time.

For more information, see [Current Locks](#).

TIP: Double-click these statistics to drill down to the [All Locks tab](#) of the Locks view.

Metrics

Seeing the current locks on the database approach 80-90% of the maximum limit allowed indicates that you should increase the Init.ora parameter dml_locks.

Troubleshooting

If the total number of locks on the database approaches the dml_locks limit, then:

- 1 Ensure that users are efficiently issuing COMMITs in transactions to release held locks before editing the Init.ora file.
- 2 Edit the Init.ora file for the database.
- 3 Increase the amount of dml_locks to a higher value.
- 4 Cycle the Oracle server when possible to allow the new value to take effect.

Current Transactions and Max Transactions

- [Metrics](#)
- [Troubleshooting](#)

This set of statistics reflects the total number of open transactions currently on the database as well as the maximum number of transactions allowed on the system at one time.

TIP: Double-click these statistics to drill down to the [Transaction Detail tab](#) of the Users Detail view.

Metrics

Seeing the current transactions on the database approach 80-90% of the maximum allowed limit allowed indicates that you should increase the Init.ora parameter transactions.

Troubleshooting

If the total number of transactions on the database approaches the transactions limit, you should:

- 1 Edit the Init.ora file for the database.
- 2 Increase the amount of transactions to a higher value.
- 3 Cycle the Oracle server when possible to allow the new value to take effect.

Total Connections and Max Connections

- [Metrics](#)
- [Troubleshooting](#)

This set of statistics reflects the total number of open sessions on the database (both active and inactive) as well as the maximum number of sessions allowed on the system at one time.

TIP: Double-click Total Connections or Max Connections to drill down to the [Sessions Overview tab](#) of the Users Detail view.

Metrics

Seeing the open connections on the database approach 80-90% of the maximum process limit allowed indicates that you should increase the process limit imposed on the database.

Troubleshooting

If the total number of connections on the database approaches the processes limit, then:

- 1 Ensure that users are efficiently logging off and on the database before editing the Init.ora file.
- 2 Edit the Init.ora file for the database.
- 3 Increase the amount of processes to a higher value.
- 4 Cycle the Oracle server when possible to allow the new value to take effect.

Inactive Connections

- [Metrics](#)
- [Troubleshooting](#)

The Inactive Connections statistic is the total number of threads logged on to the database that are currently idle.

TIP: Click this statistic to drill down to the [Sessions Overview tab](#) of the Users Detail view.

Metrics

A large number of inactive users could indicate user sessions that have mistakenly been left logged on. Because each user thread consumes a portion of memory on the Oracle server, to reduce resource usage, you should sever any session that does not need a connection.

Troubleshooting

Drill down into the Sessions Overview tab of the Users Detail view and check sessions that have many seconds idle and/or have been logged on for very long periods of time (as indicated by the logon time column). After verifying that a session is no longer necessary, you can KILL it.

Open Cursors

- [Metrics](#)
- [Troubleshooting](#)

Open Cursors is the total number of all SQL open cursors that exist on the system. In some cases, Oracle cached cursors that have been open by PL/SQL procedures can be kept open for certain lengths of time, even though the actual activity has ceased.

TIP: Click this statistic to drill down to the [Open Cursors tab](#) of the Users Detail view.

Metrics

You should monitor sessions to make sure that they do not approach the Open Cursor limit (specified in the Init.ora file). The parameter, open_cursors, limits how many open cursors (context areas) a session can have open at one time.

Troubleshooting

If the total number of open cursors approaches the open_cursors limit on Oracle8i or earlier, then:

- Ensure that user processes are efficiently using cursors before editing the Init.ora file.
- Edit the Init.ora file for the database.
- Increase the amount of open_cursors to a higher value.
- Cycle the Oracle server when possible to allow the new value to take effect.

If the total number of open cursors approaches the open_cursors limit on Oracle9i or later then:

- Change the open_cursors parameter to FORCE by using the ALTER SYSTEM SET open_cursors=< new value > command.

Users Waiting

- [Metrics](#)
- [Troubleshooting](#)

User connections that are waiting on a system generally occur for two reasons:

- 1 A process waits because a requested resource is not available.
- 2 A process waits for Oracle to perform a prerequisite task for its given operation.

Idle waits (processes waiting because they have no work) are not usually a concern. However the two wait causes mentioned above are the ones worth your time and investigation.

NOTE: This statistic displays on both the [Contention performance category view](#) and the Users performance category view.

TIP: Click this statistic to drill down to the [Session Wait Detail tab](#) of the Users Detail view.

Metrics

To determine the actual wait causes currently experienced by user connections, you should drill down from the global count of users waiting, into the actual system and user wait details of a database.

Troubleshooting

If you find a problem, drill down into wait details to determine whether the waits are resource-related.

Users Blocked

- [Metrics](#)
- [Troubleshooting](#)

A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although Oracle supports unlimited row-level locking, blocking lock situations do crop up. User processes holding exclusive locks and not releasing them via a proper COMMIT generally cause most blocks.

TIP: Click this statistic to drill down to the [Blocking Locks tab](#) of the Locks view.

Metrics

You should immediately investigate any indicator above zero for a blocking lock statistic before the situation has a chance to grow out of control.

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied. You can issue a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Performance Center makes it easier to discover the blocked lock situation, but the trick is to prevent the blocking lock situation in the first place.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT ... FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation, and know all too well that good coding practice is important. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Sessions Involved in Disk Sorts

- [Metrics](#)
- [Troubleshooting](#)

Oracle's SGA is not the only memory structure used by Oracle for database work. One of the other memory areas used by Oracle8i and earlier for normal activity is an area set aside for sort actions. When a sort operation occurs, Oracle attempts to perform the sort in a memory space that exists at the operating system level. If the sort is too large to be contained within this space, it continues the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE. Oracle records the overall number of sorts that are satisfied in memory as well as those that end up being finalized on disk. Using these numbers, you can calculate the percentage of memory sorts vs. disk sorts and get a feel for how fast your sort activity is being resolved. Obviously, memory sorts completes many times faster than sorts forced to use physical I/O to accomplish the task at hand.

Oracle9i or later now has the option of running automatic PGA memory management. Oracle has introduced a new Oracle parameter called `pga_aggregate_target`. When the `pga_aggregate_target` parameter is set and you are using dedicated Oracle connections, Oracle ignores all of the PGA parameters in the Oracle file, including `sort_area_size`, `hash_area_size` and `sort_area_retained_size`. Oracle recommends that the value of `pga_aggregate_target` be set to the amount of remaining memory (less a 10% overhead for other server tasks) on a server after the instance has been started.

TIP: Click this statistic to drill down to the [Disk Sort Detail tab](#) of the Users Detail view.

Metrics

Consistently seeing non-zero numbers for this statistic (as well as low values observed for the memory sort ratio) can indicate excessive disk sort activity. If you are on Oracle8i or earlier, increase the parameters devoted to memory sorts - `sort_area_size` and `sort_area_retained_size`.

If you are using Oracle9i or later, investigate the use of `pga_aggregate_target`. Once the `pga_aggregate_target` has been set, Oracle automatically manages PGA memory allocation, based upon the individual needs of each Oracle connection. Oracle9i or later allows the `pga_aggregate_target` parameter to be modified at the instance level with the `alter system` command, thereby lets you dynamically adjust the total RAM region available to Oracle9i.

Oracle9i also introduces a new parameter called `workarea_size_policy`. When this parameter is set to `automatic`, all Oracle connections benefit from the shared PGA memory. When `workarea_size_policy` is set to `manual`, connections allocate memory according to the values for the `sort_area_size` parameter. Under the `automatic` mode, Oracle tries to maximize the number of work areas that are using optimal memory and uses one-pass memory for the others.

Troubleshooting

If you find a problem, do the following:

- Edit the `Init.ora` or `SPFILE` file for the database.
- Increase the amount of `sort_area_size` to a higher value (take care not to not over-allocate; ensure enough free memory exists on server before increasing value). Realize that EVERY user receives this amount for sorting).
- Cycle the Oracle server when possible to allow the new value to take effect.
- Monitor new value to see if performance improves.

In addition to increasing the amount of memory devoted to sorting, you should also locate inefficient SQL that causes needless sorts. For example, in an SQL query (to eliminate duplicate rows) `UNION ALL` does not cause a sort whereas `UNION` does. People frequently code `DISTINCT` inappropriately (especially people transferring from Microsoft Access, which uses `DISTINCT` for most `SELECT` queries).

There are times you simply cannot stop sort activity. When this happens, you should try to keep it in memory whenever possible. However, large data warehousing systems frequently exhaust RAM sort allotments, so if disk sorts must occur, ensure three things:

- Your user's `TEMPORARY TABLESPACE` assignment is not the `SYSTEM` tablespace, which is the default assignment. In Oracle9i or later, you can specify a default tablespace other than `SYSTEM` for every user account that is created.
- The `TEMPORARY TABLESPACE` assigned to your users is placed on a fast disk.
- The `TEMPORARY TABLESPACE` has the tablespace parameter `TEMPORARY` assigned to it, which allows sort activity to be performed in a more efficient manner.

Leading Sessions - CPU

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Often, user connections can get out of hand with CPU use, and extreme cases have caused headaches at both the database and operating system levels.

The leading CPU session's display allows you to see who runs the leading sessions in your system with respect to CPU usage.

Metrics

Finding one or two users who use the majority of the CPU can indicate runaway or improper processes. By drilling down into the CPU activity of all users, you can quickly see if this is the case.

Leading Sessions - Memory

- [Metrics](#)
- [Troubleshooting](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problems could include a runaway process, an untuned batch procedure, or some other user-initiated operation. Frequently, user connections can get out of hand with memory consumption, with extreme cases causing headaches at both the database and operating system levels (ORA-4030 errors).

NOTE: This statistic displays on both the Users performance category view and the [Memory performance category view](#).

TIP: Click this statistic to drill down to the [Leading Sessions tab](#) of the Memory Detail view.

Metrics

If your database server does not have an overabundance of memory, you should periodically check to see who your heavy memory users are and the total percentage of memory they take. If you see one or two users who have more than 5-15% of the total memory usage, you should investigate the sessions further to see what activities they are performing.

Troubleshooting

The Leading Sessions - Memory statistic lets you find the users with the greatest allocations of overall memory. You can drill down into the details to discover the memory components that make up each user's session. The table below describes areas where you should pay particular attention to the amounts of memory usage:

Area	Description
PGA	The Program Global Area is a private memory area devoted to housing the global variables and data structures for a single Oracle process.
Memory Sorts	Contains a count of how many memory sorts a session has performed.
UGA	The User Global Area contains session-specific information regarding open cursors, state information for packages, database link information, and more.

TIP: When using Oracle's Multi-threaded Server (MTS), the UGA can be moved up into the SGA.

Leading Sessions - I/O

- [Metrics](#)

When a system undergoes heavy I/O activity, you find that all the user connections are contributing somewhat equally to the overall load. Often however, one or two user connections are responsible for 75% or more of the I/O activity. It can be that a large batch load or another typical process is running that is perfectly okay for your system or there can be a runaway process or rogue connection that you need to track down and possibly eliminate.

The Leading I/O Session statistic displays, with respect to I/O, the leading sessions in your system.

NOTE: This statistic displays on both the Users performance category view and the [I/O performance category view](#).

TIP: Click this statistic to drill down to the [Leading Sessions tab](#) of the I/O Detail view.

Metrics

Finding one or more users who are consuming more than 75% of the total I/O load can indicate a runaway or improper process. By drilling down into the I/O activity of all users, you can quickly see if this is the case.

Users Detail

The following tabbed pages are available on the Users Detail page:

- [Disk Sort Detail](#)
- [Open Cursors](#)
- [Session Waits](#)
- [Sessions Overview](#)
- [System Tablespace](#)
- [Transaction Detail](#)

Disk Sort Detail Tab

- [Metrics](#)

Something that can degrade a user's or overall database performance is disk sort activity. When a sort operation occurs, Oracle attempts to perform the sort in a memory space, assigned by the DBA, which exists at the operating system level. If the sort is too large to be contained within this space, it continues the sort on disk - specifically, in the user's assigned TEMPORARY TABLESPACE.

The Disk Sort Detail tab of the Users Detail view displays information relating to active disk sorts on a system. The table below describes the information available on the Disk Sort Detail tab of the Users Detail view:

Column	Description
User	The user account a session is using.
SID	The unique Oracle identifier for the session.
Tablespace	The tablespace being used for the disk sort.
Contents	Whether the tablespace has been set to PERMANENT or TEMPORARY.
Extents	The number of extents involved in the sort.
Blocks	The number of blocks involved in the sort.
Bytes	The total number of bytes used in the sort.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Techniques to include in your overall performance strategy are those that relate to minimizing the amount of sort activity overall and specifically sort activity that takes place on disk. A good place to start is by understanding what things cause sorts in the first place. A list of sort-related commands and SQL-related options include:

- CREATE INDEX, ALTER INDEX...REBUILD
- DISTINCT
- ORDER BY
- GROUP BY
- UNION
- INTERSECT
- MINUS
- IN, NOT IN
- Certain unindexed joins
- Certain correlated subqueries

All of these SQL commands have the potential to create a sort. As a DBA, you probably won't know which query sorts perform entirely in memory and which ones go to disk.

There are times you simply cannot stop disk sort activity (such as in data warehousing environments). That being the case, you should ensure the following are true in your database:

- Your users' TEMPORARY TABLESPACE assignment is not the SYSTEM tablespace, which is the default assignment.
- The TEMPORARY TABLESPACE assigned to your users is placed on a fast disk.
- The TEMPORARY TABLESPACE has the tablespace parameter TEMPORARY assigned to it, which allows sort activity to be performed in a more efficient manner.

If your overall memory sort ratio falls below 90%, increase the parameters devoted to memory sorts - `sort_area_size` and `sort_area_retained_size`. Keep in mind that Individual users might have the ability to alter their own sessions and increase their `sort_area_size` assignments. As a DBA, restrict users that have the ALTER SESSION privilege.

In addition to increasing the amount of memory devoted to sorting, you should also locate inefficient SQL that causes needless sorts. For example, in an SQL query (to eliminate duplicate rows) UNION ALL does not cause a sort whereas UNION does. People frequently code DISTINCT inappropriately (especially people transferring from Microsoft Access, which uses DISTINCT for most SELECT queries).

Open Cursors Tab

- [Metrics](#)

Programs opening cursors consume Oracle resources and have the potential to degrade performance, especially if their SQL code is inefficient. The Open Cursors tab of the Users Detail view allows you to quickly spot user accounts that have many cursors opened as well as the actual performance statistics for each opened cursor. The [first grid](#) displays a summary of the open cursors. The [second grid](#) displays the details of open cursors.

The table below describes the information available in the Open Cursor Summary grid on the Open Cursors tab of the Users Detail view:

Column	Description
User	The user account owning the cursors.
SID	The unique Oracle identifier for the session.
Open Cursors	The number of open cursors for the session.

The table below describes the information available in the Open Cursor Detail grid on the Open Cursors tab of the Users Detail view:

Column	Description
SID	The unique Oracle identifier for the session.
SQL Text	The SQL text being used for the cursor.
Disk reads	The number of physical I/O reads.
Buffer gets	The number of buffer gets.
Rows	The number of rows returned for the statement.
Parse Calls	The total of all parse calls to all the child cursors under this parent.
Sharable Memory	The sum of all sharable memory of all child cursors under this parent, in bytes.
Persistent Memory	The sum of all persistent memory of all child cursors under this parent, in bytes.
Runtime Memory	The sum of all ephemeral frame sizes of all children.
Sorts	The sum of sorts that were done for all the children of the parent code.
Loaded Versions	The number of children that are present in the cache and have their context heap loaded.
Loads	The number of times the object was loaded or reloaded.
Executions	The number of times the code was executed since being brought into the shared pool.
First Load Time	The creation timestamp of the parent code.

TIP: To configure a grid to display row numbers, use the [Options Editor](#).

Metrics

The Init.ora parameter OPEN_CURSORS controls the maximum number of open cursors (context areas) a session can have at one time. Seeing individual sessions approaching this limit should concern you. To avoid cursor allocation errors, set it to a very high number. The cursor limit for the Init.ora file was 255, but this limit has been increased in Oracle8.

With respect to individual cursor performance statistics, be on the lookout for cursors with abnormally high physical I/O counts (which can indicate inefficient SQL or indexing schemes), and cursors with high load counts (which can indicate an undersized library cache).

Session Waits Tab

- [Metrics](#)

Session contention is merely a subset of contention that is viewable at the global database level. Often, it takes analysis at the session level to pinpoint the exact source of contention that is occurring globally. Therefore, you need to become accustomed to viewing contention statistics and waits at the user process level.

When monitoring waits with respect to user sessions, there are two areas of interest:

- What HAS the user session been waiting on?
- What IS the user session waiting on?

Oracle records both sets of wait statistics for you. In reviewing previous waits for a session, you can see what types of things have caused a bottleneck in the session. The table below describes the information available on the Sessions Waits tab of the Users Detail view:

Column	Description
User	The user account being used by the session.
SID	The unique Oracle identifier for the session.
Wait Cause	The name of the wait cause
Program	The program being executed against Oracle by the user session.
Time (sec)	The number of seconds that process has been waiting.
Wait State	The value of the wait state. WAITING indicates that the session is waiting. WAITED UNKNOWN TIME indicates that the duration of last wait is unknown. WAITED SHORT TIME indicates that the last wait was less than 1/100th of a second. WAITED KNOWN TIME indicates that the wait is equal to the time of the last wait.
Object Owner	The name of the object owner.
Object Name	The name of the object.

NOTE: This information is available on both the [Session Waits tab](#) of the Contention Detail view and the Session Waits tab of the Users Detail view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

To view the wait times for sessions and overall system wait events, you must set the TIMED_STATISTICS parameter to TRUE for your Oracle databases. You can either set this parameter in your Init.ora file or alter the system dynamically with the ALTER SYSTEM SET TIMED_STATISTICS=TRUE command.

Some waits can be ignored like SQL*Net more data from client and SQL*Net message from client. Others like enqueue waits can be indicative of a lock contention problem. Waits, like db file scattered reads, can indicate sessions involved in long table scan operations.

Sessions Overview Tab

- [Metrics](#)

You can get more detail into dips in the overall buffer cache hit ratio by examining user I/O activity and users' individual cache hit ratios. Frequently you can pinpoint one or more accounts responsible for reducing performance and examine their SQL and other statistics to fix the situation.

The table below describes the information available on the Sessions Overview tab of the Users Detail view:

Column	Description
User	The logon name the session is using.
SID	The unique Oracle identifier for the session.
Serial #	The serial number assigned to the session.
Status	The status of the session, ACTIVE or INACTIVE.
Machine	The name of the client machine that the session is using.
O/S ID	The unique operating system identifier for the target session.
Logon Time	The date/timestamp of when the session first logged on.
Blocked	Whether the session is blocked from doing work by another session.
Seconds Idle	The number of seconds since the last time the session actively performed work.
Hit Ratio	The percentage of times the session obtained data from memory vs. physical I/O activity. The maximum value is 100%.
Program	The program being executed against Oracle by the user session.

NOTE: This information is available on both the [Sessions Overview tab](#) of the Memory Detail view and the Sessions Overview tab of the Users Detail view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Observing a poor hit ratio for one or more threads can help you pinpoint sessions that are dragging overall database performance down. Obtaining the current SQL for those sessions could yield information into the exact nature of the problem. Typically, you should investigate hit ratios under 85%. Another critical piece of information is sessions blocked. You should investigate these statistics using the Performance Center Locks view. Two other interesting pieces of information include the logon time and seconds idle. Sessions with high idle times and logon time stamps that are many hours or days old could be candidates for removal from the system.

System Tablespace Tab

- [Metrics](#)

Of all your tablespaces, you want to avoid fragmentation problems in your SYSTEM tablespace the most, as this is the major hotbed tablespace for Oracle activities. The easiest way to avoid this is to not allow any user (even the default DBA ID's SYS and SYSTEM) to have access to it. There are three ways to do this:

- 1 Ensure no user has a DEFAULT or TEMPORARY tablespace assignment of SYSTEM.
- 2 Ensure no user has a quota set for SYSTEM.
- 3 Ensure no user has been granted the UNLIMITED TABLESPACE privilege.

TIP: The System Tablespace tab of the Users Detail view identifies users that have privileges on the SYSTEM tablespace.

Metrics

Seeing any user with SYSTEM tablespace privileges should be cause for alarm. To rectify the situation, you can do one of the following:

- Use the ALTER command to change a user's DEFAULT or TEMPORARY tablespace assignment from SYSTEM to a user-defined tablespace.
- Remove any system tablespace quotas assigned to users.
- Revoke the UNLIMITED TABLESPACE privilege from any identified user.

Transaction Detail Tab

- [Metrics](#)

Long running transactions have the potential to consume large amounts of resources and cause lock contention headaches. The Transaction Detail tab of the Users Detail view provides a snapshot of transactions in progress. The table below describes the information available on the Transaction Detail tab of the Users Detail view:

Column	Description
User	The user account used by the session.
SID	The unique Oracle identifier for the session.
Rollback Segment	The name of the rollback segment.
Machine	The client machine name the session is using.
Program	The program being executed against Oracle by the session.
Status	The current status of the transaction.
Start Time	The beginning timestamp for the transaction.
Logical	The total logical I/O for the transaction.
Physical	The total physical I/O for the transaction.
Gets	The total consistent gets used by the transaction.
Changes	The total consistent changes produced by the transaction.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Seeing a long running, persistent transaction might indicate a session caught in a loop or other coding mess. Watching a transaction sit idle could clue you into a lock contention problem that exists.

Session Details

The following tabbed pages are available on the Session Detail view for Oracle:

- [Session Contention](#)
- [Session I/O](#)
- [Session Memory](#)
- [Session Network](#)
- [Session Objects](#)
- [Session SQL](#)
- [Session Statistics](#)

Session Memory Tab for Oracle

- [Metrics](#)

The Session Memory tab of the Session Detail view presents the statistics surrounding a session's memory usage. The table below describes the information available on the Session Memory tab of the Session Detail view for Oracle:

Column	Description
Statistic	The name of the memory related statistic.
Value	The cumulative value for the memory statistic.
Cache Hit Ratio	The percentage of data obtained from memory access vs. physical I/O.

Metrics

Sessions with abnormally high memory usage can affect overall performance at the server level, as this memory (PGA and UGA) is allocated outside of the Oracle SGA (unless the multi-threaded server option is being used). If cache hit ratios at the session level are lower than 85 percent, data access can be inefficient.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session I/O Tab for Oracle

- [Metrics](#)

The Session I/O tab of the Session Detail view presents the statistics surrounding a session's I/O activity. The table below describes the information available on the Session I/O tab of the Session Detail view for Oracle:

Column	Description
Statistic	The name of the I/O related statistic.
Value	The cumulative value for the I/O statistic.

Metrics

Seeing high values for physical reads and writes can indicate an inefficient session. Large numbers of physical reads can imply a session with too many large table scans or inefficient SQL operations. Large numbers of physical writes can be okay for sessions inputting large volumes of data into the database. Or, they could also indicate a session involved in heavy disk sort activity.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session Contention Tab for Oracle

- [Metrics](#)

The Session Contention tab of the Session Detail view presents information relating to resources on which the current session is waiting. The [first grid](#) displays user waits. The [second grid](#) displays user locks. The table below describes the information available in the User Waits grid on the Session Contention tab of the Session Detail view for Oracle:

Column	Description
Wait Cause	The wait event being experienced by the session.
Program	The program the session is executing against Oracle.
Seconds	The number of seconds the session has spent in the wait.
State	The status of the wait event (WAITING, etc.)

The table below describes the information available in the User Locks grid on the Session I/O tab of the Session Detail view for Oracle:

Column	Description
User	The user account being used by the session.
Terminal	The client machine name used by the session.
SID	The unique Oracle identifier for the session.
Serial #	The serial number for the session.
Table	The object locked by the session.
Lock Mode	The lock mode used by the session.
Title	The lock request issued by the session.

Metrics

You can ignore some waits, like the SQL*Net more data from client and SQL*Net message from client. Others, like enqueue waits, can be indicative of a lock contention problem. Waits, like db file scattered reads, can indicate sessions involved in long table scan operations.

Locks that are held for unusually long periods require further investigation. The application logic can be inefficient or the program is not issuing COMMITs frequently enough. The culprit of blocking-lock scenarios is usually the application design, or the SQL used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the correct SQL to get the job done is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT...FOR UPDATE statements. These place unnecessary restrictive locks on nearly every read operation. The key to avoiding lock contention is to process user transactions as quickly and efficiently as possible.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session Objects Tab for Oracle

- [Metrics](#)

The Session Objects tab of the Session Detail view presents information regarding the objects in use by a session. The table below describes the information available in the Objects Accessed grid on the Session Objects tab of the Session Detail view for Oracle:

Column	Description
Owner	The owner of the object.
Type	The type of object (TABLE, VIEW, etc.).
Object	The name of the object.

The Rollback Segments Accessed grid displays the names of rollback segments currently being used by the session.

Metrics

Once you have an idea of which objects your users access most often, you can refine some processes to facilitate access to them. You can use the Oracle 8 concept of the KEEP buffer cache to force Oracle to hold often-referenced data for data objects. The KEEP buffer cache is ideal for holding small look-up tables. If you have an earlier version of Oracle or do not want to split up your current buffer cache, you can use the CACHE table attribute to encourage Oracle to keep data blocks of CACHE'd tables at the most recently used end of the LRU buffer cache chain.

If you consistently see a session with active rollbacks, it can indicate locks are being held for long durations. It can also indicate that a session is using code without frequent enough COMMIT points.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session Network Tab for Oracle

- [Metrics](#)

The Session Network tab of the Session Detail view presents information about requests being sent to and from the database with respect to the current session. The table below describes the information available on the Session Network tab of the Session Detail view for Oracle:

Column	Description
Statistic	The name of the SQL*Net related statistic.
Value	The cumulative value of the SQL*Net related statistic.

Metrics

None.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session SQL Tab for Oracle

- [Metrics](#)

The Session SQL tab of the Session Detail view presents information relating to any current SQL issued by the session as well as any SQL previously issued by the session.

Metrics

To determine access paths, you should export and run through an EXPLAIN PLAN session any SQL that you suspect of inefficient access.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session Statistics Tab for Oracle

- [Metrics](#)

The Session Statistics tab of the Session Detail view presents information relating to all recorded performance and miscellaneous statistics for the current session. The table below describes the information available on the Session Statistics tab of the Session Detail view for Oracle:

Column	Description
Statistic	The name of the session related statistic.
Value	The cumulative value of the session related statistic.

Metrics

None.

NOTE: To configure the grid to show/hide row numbers, use the [Options Editor](#).

DB2 Expert Guide

This section includes expert help for all DB2 UDB LUW categories and statistics in the Performance Center views and pages. For detailed information on using the application, see [Using Performance Center](#). This help is divided into the following sections:

- [Home View Statistics](#)
- [Memory Page Statistics](#)
- [Object Page Statistics](#)
- [Contention Page Statistics](#)
- [I/O Page Statistics](#)
- [Users Page Statistics](#)
- [Space Page Statistics](#)
- [Instance Page Statistics](#)

Home View Statistics

In the Performance Center Home view you can review availability and overall performance of all monitored databases from a single window. Statistics on the Home view are organized into the following categories:

- [Memory Vital Signs](#)
- [Contention Vital Signs](#)
- [I/O Vital Signs](#)
- [Users Vital Signs](#)
- [Space Vital Signs](#)
- [Instance Vital Signs](#)

Memory Vital Signs

The following memory statistics are on the DB2 Home view:

- [Buffer Pool Hit Ratio](#)
- [Catalog Cache Hit Ratio](#)
- [Package Cache Overflows](#)
- [Shared Workspace Hit Ratio](#)

NOTE: For the complete set of available memory statistics, see [Memory Page Statistics](#).

Contention Vital Signs

The following contention statistics are on the DB2 Home view:

- [Deadlocks](#)
- [Lock Timeouts](#)
- [Lock Escalations](#)
- [Applications Waiting on Locks](#)

NOTE: For the complete set of available Contention statistics, see [Contention Page Statistics](#).

I/O Vital Signs

The following I/O statistics are located on the DB2 Home view:

- [Logical Read Ratio](#)
- [Physical Read Ratio](#)
- [Synchronous Read Ratio](#)
- [Asynchronous Read Ratio](#)

NOTE: For the complete set of available I/O statistics, see [I/O Page Statistics](#).

Users Vital Signs

The following users statistics are located on the DB2 Home view:

- [Connections](#)
- [Max. Connections](#)
- [Percent Executing](#)
- [Percent of Maximum](#)

NOTE: For the complete set of available Users statistics, see [Users Page Statistics](#).

Space Vital Signs

The following space statistics are located on the DB2 Home view:

- [DMS Utilization](#)
- [SMS Utilization](#)
- [Active Log Utilization](#)
- [Tablespaces Low on Space](#)

NOTE: For the complete set of available Space statistics, see [Space Page Statistics](#).

Instance Vital Signs

The following instance statistics are located on the DB2 Instance view:

- [Agent Utilization](#)

- [Sort Heap Utilization](#)
- [Percent of Agents Stolen](#)
- [Waiting for Tokens](#)

NOTE: For the complete set of available Instance statistics, see [Instance Page Statistics](#).

Memory Page Statistics

The Memory view includes the following statistics:

- [Buffer Pool Hit Ratio](#)
- [Buffer Pool Index Hit Ratio](#)
- [Database Heap Utilization](#)
- [Catalog Cache Hit Ratio](#)
- Memory Overflows
 - [Sort Overflow Percentage](#)
 - [Hash Join Overflow Percentage](#)
 - [Catalog Cache Overflows](#)
 - [Package Cache Overflows](#)
 - [Private Workspace Overflows](#)
 - [Shared Workspace Overflows](#)
- [Lock List Utilization](#)
- [Package Cache Hit Ratio](#)
- [Session Leaders – Memory](#)
- [Sort Heap Utilization](#)
- [Shared Workspace Hit Ratio](#)

Buffer Pool Hit Ratio

The DB2 database server reads and updates all data from a bufferpool because memory access is much faster than disk access. Data is copied from disk to a bufferpool as needed by the applications using the database. When the server needs to read/write data and the data is already in the bufferpool, no disk access is required. However, if the data is not in the bufferpool, it needs to be read from the disk, which is a significantly slower process.

The buffer pool hit ratio indicates the percentage of time that the database server did not need to load a page to service the request for a page. The calculation takes into account all the index and data pages that were requested.

Metrics

Avoiding disk I/O is the main issue when you try to improve the performance tuning. A high buffer pool hit ratio is desirable because it indicates a lower frequency of synchronous disk I/O. A database where data is accessed uniformly from very large tables will have a poor hit ratio. There is little you can do to improve the performance in such cases.

Troubleshooting

The buffer pool hit ratio on the database Home page is the overall hit ratio of the database. First, drill down to the bufferpool level and check the individual buffer pool hit ratios to identify the elements that may require tuning. Increasing the bufferpool size generally improves the buffer pool hit ratio. You can use one of the following methods to increase the bufferpool size depending on the layout of the data.

- 1 If the tablespaces using the bufferpool have tables and indexes on them, increase the bufferpool size incrementally until the index page hit ratio stops increasing for that bufferpool. You will not usually get any benefit increasing the bufferpool size after you reach that point.
- 2 If the tablespaces associated with the bufferpool have only *indexes* on them, increase the bufferpool size until the index page hit ratio stops increasing.
- 3 If the tablespaces associated with the bufferpool have only *data* on them, increase the bufferpool size until the data page hit ratio stops increasing.

One general design guideline is to try and use different tablespaces for indexes and tables and associate different bufferpools with them. This generally improves performance and makes bufferpool tuning much easier.

Buffer Pool Index Hit Ratio

The buffer pool index hit ratio is calculated by separating the pages and indexes that are cached by the buffer pool: $(1 - ((\text{pool_index_p_reads}) / (\text{pool_index_l_reads}))) * 100\%$.

Troubleshooting

By tuning the buffer pool index hit ratio separately, you may be able to improve the buffer pool hit ratio.

Database Heap Utilization

There is one database heap per database, and the database manager uses it on behalf of all applications connected to the database. The database heap utilization is the percentage of database heap that is currently being used.

Metrics

The database heap contains control block information for tables, indexes, table spaces, and bufferpools. It also contains space for event monitor buffers, the log buffer, and temporary memory used by utilities. Thus, the heap utilization can increase or decrease whenever any of the constituent elements change. If the utilization goes above 85% several times or stays above 85% for a sustained period, it may mean that you need to increase the maximum database heap size.

Troubleshooting

The *dbheap* database configuration parameter determines the maximum amount of heap memory that can be allocated. Examine the individual memory pools and how they are using the memory before increasing this parameter. An excessively large use of one of the memory pools may cause this problem.

Catalog Cache Hit Ratio

Catalog cache is used to cache the following types of catalog information:

- Metadata for tables, views, and aliases.
- Database authorization information that is used to check authorization when performing operations like CONNECT, LOAD, CREATE, BIND, and so on.

- Execute privileges for user-defined functions and stored procedures.

When a database operation accesses catalog information, it inserts this information into the catalog cache so operations accessing the same information can read it from the cache and avoid disk reads.

The catalog cache hit ratio indicates how well the catalog cache avoids accesses to the catalog.

Metrics

A high catalog cache hit ratio (> 80%) is desirable and it indicates that the catalog cache is working well. A smaller ratio can indicate that this parameter needs tuning. You may see a smaller ratio immediately following the first connection to the database and execution of DDL/DCL statements since these require heavy catalog accesses.

Troubleshooting

If the catalog cache hit ratio is consistently small, the database configuration parameter *catalogcache_sz* should be increased. When you increase the value of this parameter, pause to consider whether it would be more effective to allocate the extra memory you are reserving to another purpose such as package cache or bufferpools.

In a partitioned database environment, make the *catalog cache_sz* larger on the catalog partition than on other partitions because the catalog information required for all partitions will go through this partition.

When tuning this parameter, it is advisable to monitor changes to the database catalog via DDL statements. During the execution of DDL statements, there may be a large drop in the catalog cache hit ratio due to invalidation of the cached data.

Sort Overflow Percentage

The sort overflows statistic is the total number of sorts that ran out of sort heap and that may have required disk space for temporary storage. Sort Overflow Percentage is the percentage of sorts that need more heap space.

Metrics

When a sort overflows, additional overhead is incurred. The sort requires a merge phase and can potentially require more I/O if data needs to be written to disk.

Troubleshooting

Sort overflows can be reduced by increasing the *sortheap* database configuration parameter.

Hash Join Overflow Percentage

Hash join is an option for the DB2 optimizer. A hash join compares the hash codes before comparing the predicates for tables involved in a join. This reduces the number of comparisons. The hash join overflows percentage metric gives the percentage of all hash join operations that ran out of sort heap and may have required disk space for temporary storage since the current instance of Performance Analyst started monitoring the database.

Metrics

An overflowing hash join incurs a large overhead because of the amount of disk I/O required to complete the operation. If this value crosses the 30% mark, the DBA should take action.

Troubleshooting

Increase the *sortheap* database configuration parameter to reduce the number of overflows.

Catalog Cache Overflows

The catalog cache overflow statistic is the number of times the catalog cache overflowed the bounds of its allocated memory.

Metrics

Catalog cache overflows can cause unnecessary lock escalations. This can result in loss of concurrency, or 'out of memory' errors from other heaps allocated to the database's shared memory. Overflows of the catalog cache can also cause performance degradation.

DB2 reclaims the catalog cache space by evicting table descriptor information for tables, views, or aliases and/or authorization information that is not currently being used by any transaction.

Troubleshooting

Use this element with the catalog cache high watermark to determine whether the size of the catalog cache needs to be increased to avoid overflowing.

If the number of overflows is large, the catalog cache may be too small for the workload. Enlarging the catalog cache can improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures in a single unit of work, then compiling fewer SQL statements in a single transaction can improve the performance of the catalog cache. Or, if the workload includes binding packages that contain many SQL statements referencing many tables, views, aliases, user-defined functions, or stored procedures, you can try splitting packages so that they include fewer SQL statements to improve performance.

Package Cache Overflows

The package cache overflows metric is the number of times that the package cache overflowed the bounds of its allocated memory.

Metrics

Package cache overflows can cause unnecessary lock escalations. This can result in loss of concurrency, or 'out of memory' errors from other heaps allocated to the database's shared memory. Overflows of the package cache can also cause performance degradation.

Troubleshooting

Use this element with the package cache high watermark to determine whether the size of the package cache needs to be increased to avoid overflowing.

Private Workspace Overflows

The private workspace overflows statistic is the number of times that private workspaces overflowed the bounds of their allocated memory.

Metrics

Private workspace overflows can cause performance degradation as well as 'out of memory' errors from other heaps allocated to the agent's private memory.

Troubleshooting

Use this element with the private workspace high watermark to determine whether the size of the private workspace needs to be increased to avoid overflowing.

Shared Workspace Overflows

The shared workspace overflows metric is the number of times that shared workspaces overflowed the bounds of their allocated memory.

Metrics

Overflows of shared workspaces can cause performance degradation. Overflows can also 'out of memory' errors from the other heaps allocated out of application's shared memory.

Troubleshooting

Use this element with the shared workspace high watermark to determine whether the size of the shared workspaces need to be increased to avoid overflowing.

Lock List Utilization

Lock list utilization is the percentage of total database memory allocated for locks that is currently being used.

Metrics

There is only one lock list for each database and it contains the locks held by all applications connected to the database. Once the lock list is full, the database manager starts escalating row locks to table locks to free up space. This escalation may result in serious performance degradation because of reduced concurrency. Additionally, the number of deadlocks and transaction rollbacks may go up.

If this metric reaches the 75% mark, you should consider bringing this percentage down with tuning.

Troubleshooting

Depending on the database's activity level, you may be able to reduce the lock utilization by following these recommendations:

Increase size of lock list: If there is not enough lock list space available, lock escalations will occur, thereby increasing contention and reducing concurrency. Update the *locklist* database configuration parameter to increase this value.

Tune applications to reduce locking: On the Locks tab of the Users detail section, identify the applications that are holding many locks and then consider the following steps for controlling the size of the lock list:

- Make the transactions shorter by increasing the COMMIT frequency. This ensures that locks are released frequently, thus freeing up lock space.
- Before you update many rows in a single table, lock the entire table (using the SQL LOCK TABLE statement). This prevents many row-level locks from being obtained (although this decreases concurrency)
- To control how locking is done for a specific table, use the LOCKSIZE parameter of the ALTER TABLE.
- To decrease the number of share locks held, use the Cursor Stability isolation level when possible. If the applications' integrity requirements are not compromised, use Uncommitted Read instead of Cursor Stability to further decrease the amount of locking.

Decrease percentage of lock list: If a small number of applications are consuming most of the lock space, decrease the percentage of lock list for each application. You can throttle back those applications by decreasing the maxlocks database configuration parameter. This reduces the amount of lock list memory available to each application thereby allowing for better distribution of lock list memory.

NOTE: Decreasing the percentage of lock list should be the last resort, and used only if you cannot decrease utilization with the other recommendations. It can cause a large number of lock escalations.

Package Cache Hit Ratio

The sections for frequently used dynamic and static SQL statements are cached in the package cache.

The package cache hit ratio indicates how well the package cache is avoiding catalog accesses to packages and recompilations.

Metrics

A high package cache hit ratio (> 80%) is a good thing. It indicates that the package cache is working well. In the case of static SQL statements, package caching allows the Database Manager to reduce the internal overhead by eliminating the need to access system catalogs when reloading a package. For dynamic SQL, the benefit of package caching is even greater since it a cache hit eliminates the need for recompilation.

The package cache hit ratio metric is particularly important for transaction-processing applications since a typical workload involves repeatedly executing the same SQL statements.

Troubleshooting

Executing DDL statements can invalidate sections of SQL statements in the cache, causing the hit ratio to decrease dramatically. Before attempting to tune this parameter, you should check the DDL activity to see if that is causing a skew. No amount of tuning will improve the package cache performance if the DDL activity is causing package invalidations in the cache.

If the DDL activity is minimal and package cache hit ratio is consistently small, consider increasing the package cache size (`pckcachesz`) in the database configuration. When increasing the value of this parameter, consider whether it might be more effective to allocate the extra memory being reserved to another purpose such as catalog cache or bufferpools.

Session Leaders – Memory

Session Leaders – Memory shows the top five memory consuming applications that are running in the database. The details include:

Agent ID: A system-wide unique identifier for the application. On a single-partitioned database, this identifier consists of a 16 bit counter. On a multi-partitioned database, this identifier consists of the coordinating partition number concatenated with a 16 bit counter. In addition, this identifier will be the same on every partition where the application may make a secondary connection.

Auth ID: The authorization ID of the user who invoked the application that is being monitored.

Memory: The total memory pool usage for the application.

NOTE: Click the **Session Leaders -- Memory** heading to view more details on the Top Sessions View.

Sort Heap Utilization

The amount of memory allocated for each sort may be some or all of the available sort heap size. Sort heap size is the amount of memory available for each sort as defined in the database configuration parameter `sortheap`. Shared sort memory utilization gives the percentage of the sort heap being used.

It is possible for a single application to have concurrent sorts active. For example, in some cases a SELECT statement with a subquery can cause concurrent sorts.

Metrics

Memory estimates do not usually include sort heap space. If excessive sorting is occurring, the extra memory used for the sort heap should be added to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Typically the shared sort utilization should be less than or equal to 70%. You should consider tuning the database if you see a utilization value greater than this.

Troubleshooting

To bring the sort heap utilization to an acceptable level, use the following guidelines:

- Examine the queries you are running on the database to see if you can add indexes to columns frequently accessed in the *WHERE* clause. This minimizes the use of the sort heap.
- If you are using dynamic bitmaps or hash join buffers in your queries, or your queries frequently require large sorts, increase the *sortheap* size.
- If you adjust the *sortheap* size, also look at the *sheaphres* database manager configuration parameter to see if it too needs to be adjusted

If you are adjusting the *sortheap* size, you may also benefit from rebinding your packages since the optimizer takes this parameter into account when determining the access paths.

Shared Workspace Hit Ratio

When sections are required by an application for executing dynamic or static SQL statements, they are placed in the shared workspace. The shared workspace exists at the application level and is shared among applications using the database.

The hit ratio is a percentage indicating how well the shared SQL workspace is helping to avoid initialization of sections for SQL statements that are about to be executed. A high ratio indicates the shared workspace is successful in avoiding this action.

Metrics

A shared workspace is shared by many applications. If applications have similar database access patterns, they can benefit greatly if they find required sections in the shared workspace. If an application finds a section in the shared workspace (e.g., a hit), that application avoids the setup and initialization cost for that section. A high hit ratio is desirable for this metric. Typically, you should expect to see a high ratio (>80%) in transaction processing environments.

Troubleshooting

Shared workspace memory is allocated from the application control heap (*app_ctl_heap_sz* database configuration parameter) and increasing this may improve the hit ratio.

I/O Page Statistics

The I/O statistics page displays the following vital DB2 I/O statistics:

- [Direct Read Ratio](#)
- [Logical Read Ratio](#)
- [Physical Read Ratio](#)
- [Synchronous Read Ratio](#)

- [Asynchronous Read Ratio](#)
- [Direct Write Ratio](#)
- [Synchronous Write Ratio](#)
- [Asynchronous Write Ratio](#)
- [Log Read Rate](#)
- [Direct Read Rate](#)
- [Synchronous Read Rate](#)
- [Asynchronous Read Rate](#)
- [Log Write Rate](#)
- [Direct Write Rate](#)
- [Synchronous Write Rate](#)
- [Asynchronous Write Rate](#)
- I/O Agent Statistics
 - [Prefetchers](#)
 - [Page Cleaners](#)
 - [Log Space Cleans](#)
 - [Dirty Page Cleans](#)
 - [Prefetch Wait Time](#)
- Key I/O Ratios
 - [Buffer Pool Hit Ratio](#)
 - [Buffer Pool Index Hit Ratio](#)
 - [Victim Cleans Ratio](#)
 - [Threshold Cleans Ratio](#)
 - [Log Space Cleans Ratio](#)
- [Session Leaders – I/O](#)

Direct Read Ratio

Direct Read Ratio is the percentage of all reads that were direct reads. Direct reads are read operations that do not use the buffer pool.

Logical Read Ratio

Logical Read Ratio is the percentage of all reads that were logical reads. Logical Reads is the sum of all Buffer Pool Data Logical Reads and Buffer Pool Index Logical Reads.

Physical Read Ratio

Physical Read Ratio is the percentage of all reads that were physical reads. Physical Reads is the sum of all Buffer Pool Data Physical Reads and Buffer Pool Index Physical Reads.

Synchronous Read Ratio

Synchronous Read Ratio is the percentage of all index and data reads that performed synchronously by the database manager prefetchers.

Asynchronous Read Ratio

Asynchronous Read Ratio is the percentage of all index and data reads that performed asynchronously by the database manager prefetchers.

Metrics

Prefetching refers to the technique whereby the Database Manager can read several pages on the physical reads device simultaneously into a bufferpool in anticipation of access to these pages. Since each physical read operation is costly, a 50% async read ratio is desirable. It shows that the prefetchers are working well and read waits are being minimized.

Troubleshooting

The *num_ioserver* database configuration parameter defines the number of prefetchers that are available for the database. To get an optimal async read ratio, you should set the *num_ioserver* database configuration parameter to 1-2 more than the number of physical storage devices used by the database to ensure that asynchronous I/O is occurring. This parameter should be changed judiciously. Having too many prefetchers invariably results in high system I/O because prefetchers can read many more pages than required into the bufferpools.

Direct Write Ratio

Direct Write Ratio is the percentage of all writes that were direct writes. Direct Writes are write operations that do not use the buffer pool.

Synchronous Write Ratio

Synchronous Write Ratio is the percentage of all index and data writes that were performed synchronously by the database manager bufferpool page cleaners.

Asynchronous Write Ratio

Asynchronous Write Ratio is the percentage of all index and data writes that were performed asynchronously by the database manager bufferpool page cleaners.

Metrics

Page cleaners write changed pages from bufferpool to disk before the space in the bufferpool is needed by a database agent. If the page cleaners are not working well, the problem may manifest itself in two ways:

- The database agents need to synchronously free space in bufferpools resulting in poorer response time.

- If the system crashes, the recovery time of the system is greater because there will be a large number of pages that would not have been committed to disk at the time of the crash and they require processing for a large number of log records.

An overall async write ratio of 50% is desirable for most transactional database systems. If your database is 'query only' (i.e., updates are rarely or never performed), it is fine to have an async write ratio of 0 to less than 50%. Even in 'query only' databases if the queries create temp tables on the database for intermediate results, a 30-40% async write ratio is desirable.

Troubleshooting

The async write ratio can be tuned using the *num_iocleaners* database configuration parameter, which specifies the number of asynchronous page cleaners for a database. Increasing the number of page cleaners generally results in a higher async write ratio. The following rules of thumb should be followed when tuning this parameter:

- For most transactional systems set this parameter to between one and the number of physical storage devices used by the database.
- For workloads that consists mainly of simple reads from the database, set this parameter to zero.
- When you define very large bufferpools on your database you may need to increase the *num_iocleaners*
- Monitor the victim page cleaners% metric when tuning the *num_iocleaners*.

Log Read Rate

Log Read Rate is the number of pages (per second) the logger read from disk during the last monitoring interval.

Direct Read Rate

Direct Reads Rate is the number of read operations not using the buffer pool (per second) during the last monitoring interval.

Synchronous Read Rate

Synchronous Read Rate is the number of synchronous reads (per second) in the last monitoring interval.

Asynchronous Read Rate

Asynchronous Read Rate is the number of asynchronous reads (per second) in the last monitoring interval.

Log Write Rate

Log Write Rate is the number of pages (per second) the logger read from disk during the last monitoring interval.

Direct Write Rate

Direct Write Rate is the number of write operations not using the buffer pool (per second) during the last monitoring interval.

Synchronous Write Rate

Synchronous Write Rate is the number of synchronous writes (per second) in the last monitoring interval.

Asynchronous Write Rate

Asynchronous Write Rate is the number of asynchronous writes (per second) in the last monitoring interval.

Prefetchers

The *num_ioserver* database configuration parameter defines the number of prefetchers that are available for the database.

Troubleshooting

To get an optimal async read ratio, you should set the *num_ioserver* database configuration parameter to 1-2 more than the number of physical storage devices used by the database to ensure that asynchronous I/O is occurring. This parameter should be changed judiciously since having too many prefetchers invariably results in high system I/O because prefetchers can read many more pages than required into the bufferpools.

Page Cleaners

The *num_iocleaners* database configuration parameter specifies the number of asynchronous page cleaners for a database. Increasing the number of page cleaners generally results in a higher async write ratio.

Troubleshooting

The following rules of thumb can be followed when tuning this parameter:

- For most transactional systems set this parameter to between 1 and the number of physical storage devices used by the database.
- For workloads that consists of mainly simple reads from the database, set this parameter to 0.
- When you define very large bufferpools on your database you may need to increase the *num_iocleaners*.

Monitor the victim page cleaners% metric when tuning the *num_iocleaners*.

Log Space Cleans

Log Space Cleans is the number of times a page cleaner was invoked because the logging space used reached a predefined criterion for the database.

Metrics

This element can help you determine whether you have enough space for logging, and whether you need more log files or larger log files. The page cleaning criterion is set using the *softmax* configuration parameter. Page cleaners are triggered when the oldest page in the buffer pool contains an update described by a log record that is older than the current log position by the criterion value.

Dirty Page Cleans

Dirty Page Cleans is the sum of all buffer pool victim page cleaners triggered and buffer pool threshold cleaners triggered. *Buffer pool victim page cleaners triggered* is the number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database. *Buffer pool threshold cleaners triggered* is the number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database.

Prefetch Wait Time

This is the total time database applications spent waiting for prefetchers to finish loading.

Metrics

None.

Troubleshooting

If you see a high number for this metric, it indicates there is a bottleneck in the prefetcher operations. Experiment with changing the number of I/O servers (*num_ioserver*) and the I/O server sizes.

Buffer Pool Hit Ratio

The DB2 database server reads and updates all data from a bufferpool because memory access is much faster than disk access. Data is copied from disk to a bufferpool as needed by the applications using the database. When the server needs to read/write data and the data is already in the bufferpool, no disk access is required. However, if the data is not in the bufferpool, it needs to be read from the disk, which is significantly slower process.

The buffer pool hit ratio indicates the percentage of time that the database server did not need to load a page to service the request for a page. The calculation takes into account all the index and data pages that were requested.

Metrics

Avoiding disk I/O is the main issue when you try to improve the performance tuning. A high buffer pool hit ratio is desirable because it indicates a lower frequency of synchronous disk I/O. A database where data is accessed uniformly from very large tables will have a poor hit ratio. There is little you can do to improve the performance in such cases.

Troubleshooting

The buffer pool hit ratio on the database Home page is the overall hit ratio of the database. First, drill down to the bufferpool level and check the individual buffer pool hit ratios to identify the elements that may require tuning. Increasing the bufferpool size generally improves the buffer pool hit ratio. You can use one of the following methods to increase the bufferpool size depending on the layout of the data:

- 1 If the tablespaces using the bufferpool have tables and indexes on them, increase the bufferpool size incrementally until the index page hit ratio stops increasing for that bufferpool. You will not usually get any benefit increasing the bufferpool size after you reach that point.
- 2 If the tablespaces associated with the bufferpool have only *indexes* on them, increase the bufferpool size until the index page hit ratio stops increasing.
- 3 If the tablespaces associated with the bufferpool have only *data* on them, increase the bufferpool size until the data page hit ratio stops increasing.

One general design guideline is to try and use different tablespaces for indexes and tables and associate different bufferpools with them. This generally improves performance and makes bufferpool tuning much easier.

Buffer Pool Index Hit Ratio

The buffer pool index hit ratio is calculated by separating the pages and indexes that are cached by the buffer pool: $(1 - ((\text{pool_index_p_reads}) / (\text{pool_index_l_reads}))) * 100\%$.

Troubleshooting

By tuning the buffer pool index hit ratio separately, you may be able to improve the buffer pool hit ratio.

Victim Cleans Ratio

Victim Cleans Ratio is the percentage of all page cleans that were victim page cleans. Victim page cleans are triggered when a synchronous write is needed during the victim buffer replacement for the database.

Troubleshooting

If this ratio is low, you may have defined too many page cleaners. Because the buffer pool is meant to defer writing to the last possible moment, having too many page cleaners is counterproductive. If the ratio is high, it might mean that you have too few page cleaners defined, which will increase recovery time after failures.

Threshold Cleans Ratio

Threshold Cleans Ratio is the percentage of all page cleans that were invoked because a buffer pool had reached the dirty page threshold criterion for the database.

Metrics

This dirty page threshold is set by the *chnpggs_thresh* configuration parameter. Cleaners are triggered when the number of dirty pages in the pool exceeds the percentage (threshold) applied to the buffer pool size.

Troubleshooting

If this ratio is too low, pages could be written out too early, and they will need to be read back in. If the ratio is too high, too many pages may accumulate, and users will have to write out pages synchronously.

Log Space Cleans Ratio

The log space cleaners triggered metric is the number of times a page cleaner was triggered because log space usage reached a predefined threshold for the database.

Metrics

The log space cleaners are triggered every time the space used by the DB2 log reaches the limit set in the *softmax* database configuration parameter. This parameter specifies the percentage of primary log size at which the cleaners are triggered. By default this parameter is set to 100.

A low rate of log space cleaners getting triggered may indicate that logs are not being written to disk frequently enough and that you may need to process a large number of log records and redundant log records in the event of a crash recovery.

High rates of log space cleaners getting triggered can indicate that your primary log is too small or that you have set the *softmax* too high. In either case, a very high rate of log cleaners being triggered may adversely impact database performance.

It is important to keep in mind, however, that more page cleaner triggers and more frequent soft checkpoints increase the overhead associated with database logging. This can have an impact on the performance of the Database Manager. Also, more frequent soft checkpoints may not reduce the time required to restart a database, if you have:

- Very long transactions with few commit points.
- A very large bufferpool and the pages containing the committed transactions are not written back to disk very frequently. The use of asynchronous page cleaners can help avoid this situation.

In both of these cases, the log control information kept in memory does not change frequently and there is no advantage in writing the log control information to disk, unless it has changed

Troubleshooting

Examine this metric together with other page cleaner metrics and the async write ratio to determine if excessive numbers of log page cleaners are being triggered. If this is true, you need to either increase your primary log size or adjust the *softmax* parameter to a higher value.

NOTE: Decreasing the log page cleaner triggering rate can impact the overall crash recovery time so it needs to be done judiciously.

Session Leaders – I/O

Session Leaders – I/O shows the top five applications consuming the most I/O. The details include:

Application	Description
Agent ID	A system-wide unique ID for the application. On a single-partitioned database, this identifier consists of a 16 bit counter. On a multi-partitioned database, this identifier consists of the coordinating partition number concatenated with a 16 bit counter. In addition, this identifier will be the same on every partition where the application may make a secondary connection.
Auth ID	The authorization ID of the user who invoked the application that is being monitored. On a DB2 Connect gateway node, this is the user's authorization ID on the host.
Total I/O (ms)	The total amount time (in milliseconds) spent performing buffered and direct reads and writes.

Space Page Statistics

- Database Overview
 - [DMS Space \(Total Used and Total Free\)](#)
 - [DMS Utilization](#)
 - [SMS Space \(Total Used and Total Free\)](#)
 - [SMS Utilization](#)

- Log Overview
 - [Active Log Size](#)
 - [Active Log Used](#)
 - [Active Log Free](#)
 - [Active Log Utilization](#)--+
 - [Secondary Logs Allocated](#)
 - [Secondary Log Used HWM](#)
- Exception Summary
 - [Inaccessible Containers](#)
 - [Tablespaces Low on Space](#)
 - [Abnormal State Tablespaces](#)
- [Tablespace Overview](#)

DMS Space (Total Used and Total Free)

The total used and total free metrics show the space details for DMS tablespaces. Specifically, they show the amount of used and free space on the DMS tablespaces in the database.

Metrics

Unlike SMS tablespaces, the total available space to the DMS tablespaces is defined at the time the database is defined. The DBA needs to explicitly increase the amount of space on DMS tables by adding/extending tablespace containers (using the ALTER TABLESPACE statement). If left unattended, a DMS tablespace can either remain underutilized or fill up completely. Keeping an eye on the DMS space is important because the once it fills up, the applications trying to write to the database will come to a halt suddenly.

Troubleshooting

Go to the Space home page or drilldown to see the usage metrics for individual tablespaces and allocate/deallocate space to containers accordingly using ALTER TABLESPACE command.

DMS Utilization

DMS utilization is the percentage of all DMS tablespaces that are free.

SMS Space (Total Used and Total Free)

This metric shows the total amount of free and used SMS space by the database.

Metrics

The maximum size of SMS tablespaces is not set at the time the tablespaces are created. The maximum size that can be reached is the space available on the drives/volumes that the SMS tablespace containers are defined on. You need to periodically examine the available space of the drives/volumes (using OS metrics Space page) to make sure there is enough space available for your database requirements. Since the space is determined by the space available on the drives/volumes, remember that if other applications are reading and writing to the same devices (especially logs, traces, etc), DB2 may be periodically competing for space with these applications.

Troubleshooting

Ideally, try and isolate the DB2 SMS drives/volumes from other applications. If the OS level metrics show that you are running out of space on a volume that is used by a tablespace's containers, you can add more containers defined on volumes with more space, to the tablespace using the ALTER TABLESPACE command.

SMS Utilization

SMS utilization is the percentage of all SMS tablespaces that are free.

Active Log Size

Active Log Size is the total amount of active log space (in megabytes) available in the database.

Active Log Used

Active Log Used is the current amount of active log space (in megabytes) being used in the database.

Active Log Free

Active Log Free is the active log space (in megabytes) in the database that is not currently being used.

Active Log Utilization

Active Log Utilization is the percentage of space consumed in the database. A very high percentage may cause an alert.

Secondary Logs Allocated

This section shows the secondary logs allocations by the database over the monitoring period.

Metrics

When the primary log files become full, the secondary log files are allocated one at a time as needed, up to a maximum number as controlled by the logsecond database configuration parameter. Secondary log files are useful in preventing occasional log fill ups but they may not be as efficient as primary log and they also increase the recovery time. If you see a constant reliance on secondary log files, it may indicate infrequent commits by database applications or insufficient primary log space.

Troubleshooting

First examine the database applications to see if secondary log files are due to long running transactions and whether these transactions can be shortened. If the transactions cannot be shortened or you still see a very frequent use of secondary logs after transaction tuning, increase the primary log size (logprimary, logfilesiz Database configuration parameters) to reduce the dependence on secondary logs.

If there is an occasional long running transaction, and you see your see transaction log full (SQL0964C) error messages, you can either increase the number of secondary log files (logsecond database configuration parameter) or set the number to -1 (no limit on number of secondary log files).

Secondary Log Used HWM

Secondary Log Used HWM is the maximum amount of secondary log space used (in megabytes).

Troubleshooting

If this value is high, you can try several options:

- Increase the size of the log files.
- Create more primary logs.

The following configuration parameters may need to be updated as a result:

- logfilsiz
- logprimary
- logsecond
- logretain

NOTE: If there are no secondary log files defined, the value will be zero.

Inaccessible Containers

The inaccessible containers metric identifies the number of containers on all tablespaces that are currently inaccessible.

Metrics

Inaccessible containers represent a serious problem on the database. The database is unable to access/write any data on an inaccessible container. Usually this is caused by either media errors or modifications to the container files/directories from outside DB2.

Troubleshooting

Drill down to the Space metrics to examine the inaccessible containers and the tablespaces on which they are contained. You may need to perform a redirected restore on the tablespace from a backup to correct this.

Tablespaces Low on Space

Tablespaces Low on Space is the number of tablespaces in the database where the used space on the tablespace is more than 80% of the total available space.

Metrics

This metric is an indicator that some of the tablespaces in your database may be running out of space. You should drill down to identify the tablespaces that may be at the critical utilization level. In tablespaces where there are little or no inserts after the initial data is loaded, little or no free space may have been left on purpose. In such cases, it is normal to see a high utilization.

Troubleshooting

If this metric goes beyond the normal operational value for your database, consider creating additional space on the tablespaces that are running out of space. You can do this by performing one or more one of the following tasks:

- Clean up unused space by deleting the tables/indexes that are no longer required.

- Resize or extend the existing containers in the tablespaces.
- Add new containers to the tablespaces.

Abnormal State Tablespaces

The tablespaces in abnormal state metric is a count of the number of tablespaces in your database that are not in a 'Normal State'.

Metrics

Tablespaces in not normal state are not necessarily a cause for alarm. They may have been transitioned to that state intentionally by the DBA for maintenance related tasks. If this metric shows a value that is higher than you anticipated, you may need to drilldown to the Space metrics to see which tablespaces are not normal.

A tablespace can be in one or more of the following states when it is not 'Normal':

- Quiesced: SHARE, UPDATE, EXCLUSIVE
- Pending: Load, delete, backup, roll forward, restore, disable, drop
- In progress: Roll forward, reorg, backup, TBS deletion, TBS creation
- Storage must be defined
- Restore in progress
- Offline and not accessible
- Storage may be defined
- Storage definition is in 'final' state
- Storage definition was changed prior to rollforward
- DMS rebalancer is active

Tablespace Overview

The Tablespace Overview section includes a list of all of the tablespaces in the database. The following information is presented for each tablespace:

- **Tablespace:** The name of the tablespace.
- **Type:** system managed (SMS) or database managed (DMS) tablespace.
- **State:** This element describes the current state of the tablespace.
- **Size (MB):** Total space (Used and Free) used by the tablespace on the storage device(s). This will be the same as used space for SMS tables.
- **Used (MB):** The space currently in use on tablespace.
- **Free (MB):** The space currently free on the tablespace. (This is not applicable to SMS tablespaces.)
- **% Used:** The percentage of used space on the tablespace.

Object Page Statistics

The Object view includes the following statistics:

- [Buffer Pools Tab – Overview](#)
- [Buffer Pools Tab – I/O Analysis](#)
- [Tablespaces Tab – Overview](#)
- [Tablespaces Tab – I/O Analysis](#)
- [Containers Tab](#)
- [Tables Tab](#)

Buffer Pools Tab – Overview

The Buffer Pools tab displays details about the buffer pools in the database being monitored:

Buffer Pool Details	Description
Node:	The unique identification number assigned to the node.
Bufferpool:	The name of the buffer pool.
Size (MB):	The current buffer pool size.
Page Size (KB):	The default page size of the buffer pool.
Files Closed:	The number of times a database file was closed because the limit for concurrently open files was reached.
Overall Hit Ratio:	Indicates the percentage of time that the database server did not need to load a page from in order to service the request for a page.
Index Hit Ratio:	The percentage of all index reads that were satisfied because the page was already available in a bufferpool.
Avg Read Time (ms):	The average amount of time (in milliseconds) spent reading in data and index pages from table space containers.
Avg Write Time (ms):	The average amount of time (in milliseconds) spent writing data and index pages from the buffer pool to disk.
Pages Read Per Minute:	The number of physical reads per minute.

Buffer Pools Tab – I/O Analysis

The Buffer Pools tab displays details about the buffer pools in the database being monitored:

Buffer Pool Details	Description
Node:	The unique identification number assigned to the node.
Bufferpool:	The name of the buffer pool.

Buffer Pool Details	Description
% Async Reads:	The percentage of all index and data reads performed asynchronously by the database manager prefetchers.
% Sync Reads:	The percentage of all index and data reads that performed synchronously by the database manager prefetchers.
% Async Writes:	The percentage of all index and data writes performed asynchronously by the database manager bufferpool page cleaners.
% Sync Writes:	The percentage of all index and data writes that were performed synchronously by the database manager bufferpool page cleaners.
% Block I/O:	The percentage of all block/vectored I/O that is block I/O. Block I/O represents the number of times DB2 performs sequential prefetching of pages into the block area of the buffer pool.
% Vectored I/O:	The percentage of all block/vectored I/O that is vectored I/O. Vectored I/O represents the number of times DB2 performs sequential prefetching of pages into the page area of the buffer pool.
Avg Async Read (ms):	The average amount of time spent reading data and index pages.
Avg Sync Read (ms):	The average amount of time spent performing synchronous reads.
Avg Async Write (ms):	The average amount of time spent writing data and index pages to disk.
Avg Sync Write (ms):	The average amount of time spent performing synchronous writes.

Tablespaces Tab – Overview

The Tablespaces tab displays details about the tablespaces in the database being monitored:

Tablespaces	Description
Tablespace:	The name of the tablespace.
Type:	System managed (SMS) or database managed (DMS) tablespace.
State:	This element describes the current state of the tablespace.
Size (MB):	The current buffer pool size.
Used (MB):	The space currently in use on tablespace.
Free (MB):	The space currently free on the tablespace. (This is not applicable to SMS tablespaces.)
% Used:	The percentage of space currently in use on tablespace.
Page Size (KB):	The default page size of the buffer pool.

Tablespaces	Description
Files Closed:	The number of times a database file was closed because the limit for concurrently open files was reached.
Overall Hit Ratio:	Indicates the percentage of time that the database server did not need to load a page from in order to service the request for a page.
Index Hit Ratio:	The percentage of all index reads that were satisfied because the page was already available in a bufferpool.
Avg Read Time (ms):	The average amount of time (in milliseconds) spent reading in data and index pages from table space containers.
Avg Write Time (ms):	The average amount of time (in milliseconds) spent writing data and index pages from the buffer pool to disk.
Pages Read Per Minute:	The number of physical reads per minute.

Tablespaces Tab – I/O Analysis

The Tablespaces tab displays details about the tablespaces in the database being monitored:

Tablespace: The name of the tablespace.

Bufferpool: The name of the buffer pool.

% Async Reads: The percentage of all index and data reads performed asynchronously by the database manager prefetchers.

% Sync Reads: The percentage of all index and data reads that performed synchronously by the database manager prefetchers.

% Async Writes: The percentage of all index and data writes performed asynchronously by the database manager bufferpool page cleaners.

% Sync Writes: The percentage of all index and data writes that were performed synchronously by the database manager bufferpool page cleaners.

% Block I/O: The percentage of all block/vectored I/O that is block I/O. Block I/O represents the number of times DB2 performs sequential prefetching of pages into the block area of the buffer pool.

% Vectored I/O: The percentage of all block/vectored I/O that is vectored I/O. Vectored I/O represents the number of times DB2 performs sequential prefetching of pages into the page area of the buffer pool.

Avg Async Read (ms): The average amount of time spent reading data and index pages.

Avg Sync Read (ms): The average amount of time spent performing synchronous reads.

Avg Async Write (ms): The average amount of time spent writing data and index pages to disk.

Avg Sync Write (ms): The average amount of time spent performing synchronous writes.

Async Pages Read Per Request: The average number of pages read per asynchronous request.

Containers Tab

The Containers tab displays details about the containers for a tablespace:

Node: The unique identification number assigned to the node.

Tablespace: The name of the tablespace.

ID: A value that uniquely identifies a tablespace used by the current database.

Type: System managed (SMS) or database managed (DMS) tablespace.

Container: The name of the container. Typically this is the full path of the file/directory/device where the container exists.

ID: A value that uniquely identifies the container within the tablespace

Type: The type of container. An SMS Container will be a directory. A DMS Containers will be a raw device/file/stripped disk/ or stripped file. Together with Container name, and partition, this metric identifies the physical location of the container.

Accessible: This element describes if a container is accessible or not.

Size (MB): The size of the container (in megabytes).

Total Pages: Total pages in the container.

Useable Pages: Usable pages in the container (applicable to DMS Tablespaces only).

Node Name: The name of the node.

FS Type: This is the type of file system that the container is defined on such as 'NTFS'. It is obtained from the IBM supplied user defined function "SYSPROC.SNAPSHOT_CNTRFS" and is only available for DB2 v8.2 databases and above.

FS Size (GB): This is the total size in gigabytes (GB) of the file system that the container is defined on. It is obtained from the IBM supplied user defined function "SYSPROC.SNAPSHOT_CNTRFS" and is only available for DB2 v8.2 databases and above.

FS Free (GB): This is the total amount of space in gigabytes (GB) that is available on the file system that the container is defined on. It is obtained from the IBM supplied user defined function "SYSPROC.SNAPSHOT_CNTRFS" and is only available for DB2 v8.2 databases and above.

NOTE: If the DB2 database is not a DB2 v8.2 database the values in these columns will be N/A or zero.

Tables Tab

The Tables tab displays details about the tables in the database being monitored:

Schema: The schema name for the table.

Table Name: The name of the table.

Type: The type of table for which information is returned.

Tablespace: Tablespace where the table data resides.

Bufferpool: The name of the buffer pool.

Rows Read Per Tran: The average number of rows read per transaction.

Rows Read: The number of rows read from the table.

Rows Written: The number of rows written to each table.

Overflow Accesses: The number of reads and writes to overflowed rows in this table.

Page Reorgs: The number of page reorganizations executed for a table.

Data Pages (MB): The amount of tablespace used by data pages (in megabytes).

Index Pages (MB): The amount of tablespace used by index pages (in megabytes).

LOB Pages (MB): The amount of tablespace used by LOB pages (in megabytes).

Long Pages (MB): The amount of tablespace used by long pages (in megabytes).

Contention Page Statistics

The Contention view includes the following statistics:

- System Contention
 - [Deadlocks](#)
 - [Lock Waits](#)
 - [Lock Timeouts](#)
 - [Lock Escalations](#)
 - [Applications Waiting on Locks](#)
- [Lock Overview](#)
- [Lock List Utilization](#)
- [Lock Wait Time](#)
- [Average Lock Wait Time](#)

Deadlocks

Deadlocks shows the total number of deadlocks that have occurred since this instance of Performance Center started monitoring the database.

Metrics

If a large number of deadlocks are detected, it can indicate that applications are experiencing lock contention problems. Deadlocks are usually caused by one of the following situations:

- Lock escalations on the database.
- Catalog tables locked for Repeatable Read.
- Applications are using inappropriate isolation levels at bind time.
- Applications are obtaining the same locks in a different order.
- Applications are locking tables explicitly where row level locks are sufficient.

Troubleshooting

You may be able to modify the applications causing lock contentions for better concurrency.

To identify the applications that may be causing contentions, go to the Lock View.

Lock Waits

At the database level, this is the total number of times that applications have had to wait for locks within this database. At the application-connection level, this is the total number of times that this connection requested a lock but had to wait because another connection was already holding a lock on the data.

Metrics

This element may be used with `lock_wait_time` to calculate, at the database level, the average wait time for a lock. This calculation can be done at either the database or the application-connection level.

Troubleshooting

If the average lock wait time is high, you should look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, then the values of one or both of the `locklist` and `maxlocks` configuration parameters may be too low.

Lock Timeouts

The lock timeouts metric identifies the number of times that a request to lock an object timed out without being granted.

Metrics

If the number of lock timeouts becomes excessive when compared to the acceptable range for your database, it can indicate that an application is holding locks for long durations. It can also indicate that the amount of time an application waits for a lock before timing out is too short.

If you have too few lock timeouts and the average lock wait time is too high, it can indicate that the lock timeout configuration parameter is set to an excessively high value.

Troubleshooting

First you should examine the lock activity at the application level to identify any particular application that is causing excessive lock contentions. If so, you can tune the application to provide better concurrency. If lock timeouts are excessive, and average lock wait times are very short, you can increase the `locktimeout` database configuration parameter to make the applications wait longer before timing out.

Lock Escalations

The lock escalations metric indicates the number of times that locks have been escalated from row locks to table locks, since this instance of Performance Center started monitoring the database.

Metrics

A lock is escalated when the total number of locks held by an application reaches the maximum amount it is allowed in the lock list memory. There are several possible causes of lock escalations:

- The database lock list size is too small for the concurrent workload
- The maximum percentage of lock list memory allowed for each application is too small
- One or more applications are obtaining an excessive number of locks

Monitor the lock escalations over a period of time to determine what levels are acceptable in your environment. If the escalations are excessive, or are accompanied by deadlocks or long lock waits, consider tuning the database.

Applications Waiting on Locks

The Apps Waiting on Locks metric gives the percentage of all currently connected applications that are waiting for locks.

Metrics

If this number is high, you should investigate whether the applications are having concurrency problems.

Troubleshooting

Compare this metric with the lock escalations metric to identify if the lock list memory is too small.

Go to the Locks Waiting Details tab of the Lock View and examine the lock activity at application level to identify the applications that are holding a large number of row locks and table-level locks. You may be able to tune the applications with a high number of locks.

Lock Overview

Agent ID – A system-wide unique ID for the application. On a single-partitioned database, this identifier consists of a 16 bit counter. On a multi-partitioned database, this identifier consists of the coordinating partition number concatenated with a 16 bit counter. In addition, this identifier will be the same on every partition where the application may make a secondary connection.

Auth ID – The authorization ID of the user who invoked the application that is being monitored. On a DB2 Connect gateway node, this is the user's authorization ID on the host.

Application – The name of the application running at the client, as known to the database or DB2 Connect server.

Locks Waiting – The total number of times that applications or connections waited for locks.

Lock Wait Time (ms) – The total elapsed time applications waited for a lock (given in milliseconds).

Timeouts – Indicates the timeout value (in seconds) when an application has issued a SET CURRENT LOCK TIMEOUT statement. In cases where the statement has not been executed, the database level lock timeout will be shown.

Deadlocks – Shows the total number of deadlocks that have occurred since this instance of Performance Center started monitoring the database.

Lock List Utilization

Lock list utilization is the percentage of total database memory allocated for locks that is currently being used.

Metrics

There is only one lock list for each database and it contains the locks held by all applications connected to the database. Once the lock list is full, the database manager starts escalating row locks to table locks to free up space. This escalation may result in serious performance degradation because of reduced concurrency. Additionally, the number of deadlocks and transaction rollbacks may go up.

If this metric reaches the 75% mark, you should consider bringing this percentage down with tuning.

Troubleshooting

Depending on the database's activity level, you may be able to reduce the lock utilization by following these recommendations:

Increase size of lock list: If there is not enough lock list space available, lock escalations will occur, thereby increasing contention and reducing concurrency. Update the *locklist* database configuration parameter to increase this value.

Tune applications to reduce locking: On the Lock View, identify the applications that are holding many locks and then consider the following steps for controlling the size of the lock list:

- Make the transactions shorter by increasing the COMMIT frequency. This ensures that locks are released frequently, thus freeing up lock space.
- Before you update many rows in a single table, lock the entire table (using the SQL LOCK TABLE statement). This prevents many row-level locks from being obtained (although this decreases concurrency).
- To control how locking is done for a specific table, use the LOCKSIZE parameter of the ALTER TABLE.
- To decrease the number of share locks held, use the Cursor Stability isolation level when possible. If the applications' integrity requirements are not compromised, use Uncommitted Read instead of Cursor Stability to further decrease the amount of locking.

Decrease percentage of lock list: If a small number of applications are consuming most of the lock space, decrease the percentage of lock list for each application. You can throttle back those applications by decreasing the maxlocks database configuration parameter. This reduces the amount of lock list memory available to each application thereby allowing for better distribution of lock list memory.

NOTE: Decreasing the percentage of lock list should be the last resort, and used only if you cannot decrease utilization with the other recommendations. It can cause a large number of lock escalations.

Lock Wait Time

Lock Wait Time is the elapsed time (in milliseconds) that all applications were waiting for a lock.

Metrics

At the database level, this is the total amount of elapsed time that all applications were waiting for a lock within this database. At the application-connection and transaction levels, this is the total amount of elapsed time that this connection or transaction has waited for a lock to be granted to it.

This element may be used in conjunction with the lock_waits monitor element to calculate the average wait time for a lock. This calculation can be performed at either the database or the application-connection level.

When using monitor elements providing elapsed times, you should consider:

- Elapsed times are affected by system load, so the more processes you have running, the higher this elapsed time value.
- To calculate this element at the database level, the database system monitor sums the application-level times. This can result in double counting elapsed times at a database level, since more than one application process can be running at the same time.

Troubleshooting

None.

Average Lock Wait Time

This is the average time, applications waited for locks in the database.

Metrics

This metric helps you determine if the applications are spending a large amount of time waiting to obtain locks.

Troubleshooting

If the average lock wait time is high, you should look for applications that hold many locks or have lock escalations. If appropriate, you may need to tune such applications for better concurrency. If escalations are the reason for high average lock wait time, the values of one or both of the *locklist* and *maxlocks* database configuration parameters may be too low.

Users Page Statistics

The Users view includes the following statistics:

- Session Activity Summary
 - [Connections](#)
 - [Connections Idle](#)
 - [Connections Active](#)
 - [Connections Waiting](#)
 - [Static SQL Statements](#)
 - [Dynamic SQL Statements](#)
- Session Transaction Summary
 - [Transactions](#)
 - [Transactions Per Second](#)
 - [Sorts Per Transaction](#)
 - [Lock Waits Per Transaction](#)
 - [Selects Per Transaction](#)
 - [Rows Selected Per Transaction](#)
- [Session Leaders – Memory](#)
- [Session Leaders – I/O](#)
- [Session Leaders – CPU](#)
- [Max. Connections](#)
- [Percent Executing](#)
- [Percent of Maximum](#)

Related Topics

- [Users Detail](#)

Connections

The total connections metric is the number of applications currently connected to the database.

Metrics

You can use this metric to help you get an overview of the level of database activity and the amount of system resources in use.

Troubleshooting

This metric can help you adjust the setting of the *maxappls* and *max_coordagents* configuration parameters. For example, if this value is always the same as *maxappls*, you consider increasing the value of *maxappls*.

Connections Idle

The connections idle metric indicates the number of applications that are currently connected to the database for which the database manager is not executing any requests.

Metrics

You may use this element to help you understand the level of activity within a database and the amount of system resource being used.

Connections Active

The connections active statistic indicates the number of applications for which the database manager is currently executing requests.

Metrics

You can use this element to understand how many of the database manager agent tokens are being used by applications connected to this database.

Connections Waiting

This metric indicates the percentage of all connected applications waiting on locks.

Metrics

If this number is high, the application may have concurrency problems.

Troubleshooting

You can identify the applications that are holding locks or exclusive locks for a long time from the Users Detail>Locks page and tune such applications for better concurrency.

Static SQL Statements

The static SQL metric is the number of static SQL statement executions attempted on the database each second.

Dynamic SQL Statements

The dynamic SQL metric is the number of dynamic SQL statement executions being attempted on the database each second.

Transactions

Transactions is the number of transactions executed on the database since the start of this instance of Performance Center.

Metrics

A low number of units of work compared to the overall SQL activity (static + dynamic SQL statements) indicate long transactions. This may in turn be an indicator of poor concurrency and heavy log usage.

Transactions Per Second

Transactions per second is the number of transactions (units of work) completed per second on the database.

Metrics

A small rate of transactional activity on the database can indicate that applications are not doing frequent commits, which may lead to logging and concurrency problems.

Troubleshooting

Drill down to the Users Detail>SQL Activity page to check which applications are running their transactions for long periods of time.

Sorts Per Transaction

Sorts per transaction is the number sorts that have been executed during per transaction.

Lock Waits Per Transaction

Lock Waits Per transaction is the number of times per transaction applications or connections had to wait because another connection was holding a lock on the data.

Selects Per Transaction

Selects per transaction is the number of SQL SELECT statements executed per transaction.

Rows Selected Per Transaction

Rows selected per transaction is the number of rows per transaction that have been selected and returned to the application.

Session Leaders – Memory

Session Leaders – Memory shows the top five memory consumption applications that are running in the database. The details include:

Session Leader Details	Description
Agent ID:	A system-wide unique ID for the application. On a single-partitioned database, this identifier consists of a 16 bit counter. On a multi-partitioned database, this identifier consists of the coordinating partition number concatenated with a 16 bit counter. In addition, this identifier will be the same on every partition where the application may make a secondary connection.
Auth ID:	The authorization ID of the user who invoked the application that is being monitored. On a DB2 Connect gateway node, this is the user's authorization ID on the host.
Memory:	Total memory pool usage for the application.

Session Leaders – I/O

I/O shows the top five applications consuming the most I/O. The details include:

Details	Description
Agent ID	A system-wide unique ID for the application. On a single-partitioned database, this identifier consists of a 16 bit counter. On a multi-partitioned database, this identifier consists of the coordinating partition number concatenated with a 16 bit counter. In addition, this identifier will be the same on every partition where the application may make a secondary connection.
Auth ID	The authorization ID of the user who invoked the application that is being monitored. On a DB2 Connect gateway node, this is the user's authorization ID on the host.
Total I/O (ms)	The total amount time (in milliseconds) spent performing buffered and direct reads and writes.

Session Leaders – CPU

Session Leaders – CPU shows the top five applications with the highest CPU usage. The details include:

Details	Description
Agent ID:	A system-wide unique ID for the application. On a single-partitioned database, this identifier consists of a 16 bit counter. On a multi-partitioned database, this identifier consists of the coordinating partition number concatenated with a 16 bit counter. In addition, this identifier will be the same on every partition where the application may make a secondary connection.

Details	Description
Auth ID	The authorization ID of the user who invoked the application that is being monitored. On a DB2 Connect gateway node, this is the user's authorization ID on the host.
CPU (sec):	The total user and system CPU time used by the application agents.

Max. Connections

This parameter (*maxappls*) is the maximum number of concurrent applications that can be connected locally and remotely to a database. Allowing a higher number of concurrent applications to connect increases the potential for more memory use.

Metrics

If you set *maxappls* to *automatic*, any number of applications will be allowed to connect. Otherwise, you can set the *maxappls* parameter to a specific number that allows for all of the following:

- The sum of the connected applications.
- The number of these connected applications that might be concurrently in the process of completing a two-phase commit or rollback.
- The expected number of indoubt transactions that may exist at any given time.

Percent Executing

Percent Executing is the percentage of applications that are currently executing in the database, and for which the database manager is currently processing a request.

Percent of Maximum

Percent of Maximum is the percentage of the maximum number of allowed concurrent applications that are currently connected to the database.

Users Detail

The following tabbed pages are available on the Users Detail page:

Overview Tab

Column	Description
Agent ID	This is a system-wide unique identifier for the application.
Auth ID	The authorization ID of the user who invoked the application that is being monitored.
Client PID	The process ID of the client application that made the connection to the database.
OS User ID	The authorization ID used to access the operating system.
Application	Name of the application executable.
Status	The current status of the application.
Coord Node	In a multi-node system, this is the node number of the node where the selected application connected or attached to the instance.
UOW Elapsed Time (sec)	The elapsed execution time of the most recently completed unit of work.
Memory Used (KB)	Total memory pool usage for the application.
Lock Wait Time (ms)	The current amount of wait time for the process, in milliseconds.
Total Sort Time (ms)	The total elapsed time (in milliseconds) for all sorts that have been executed.
User CPU Time (sec)	The total user CPU time used by the application agents.
System CPU Time (sec)	The total system CPU time used by the application agents.
Buffered I/O Time (ms)	The total time spent by application in performing buffered reads and writes.
Direct I/O Time (ms)	The total time spent by application in performing non-buffered reads and writes.

Locking Tab

Column	Description
Agent ID	This is a system-wide unique identifier for the application.
Auth ID	The authorization ID of the user who invoked the application that is being monitored.
Client PID	The process ID of the client application that made the connection to the database.
OS User ID	The authorization ID used to access the operating system.
Application	Name of the application executable.
Status	The current status of the application.
Coord Node	In a multi-node system, this is the node number of the node where the selected application connected or attached to the instance.
Lock Wait Time (ms)	The current amount of wait time for the process, in milliseconds.
Lock Waits	At the database level, this is the total number of times that applications have had to wait for locks within this database. At the application-connection level, this is the total number of times that this connection requested a lock but had to wait because another connection was already holding a lock on the data.
Locks Waiting	The number of agents waiting on a lock.
Locks Held	The number of locks currently held either by all applications in the database (database level) or by all agents for the application (application level).
Deadlocks	The total number of deadlocks that have occurred since this instance of Performance Center started monitoring the database.

Column	Description
Timeouts	The number of times that a request to lock an object timed out without being granted.
Timeout Value (sec)	Indicates the timeout value (in seconds) when an application has issued a SET CURRENT LOCK TIMEOUT statement.
Lock Escalations	The number of times that locks have been escalated from row locks to table locks, since this instance of Performance Center started monitoring the database.
X Lock Escalations	The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.

Sorting Tab

Column	Description
Agent ID	This is a system-wide unique identifier for the application.
Auth ID	The authorization ID of the user who invoked the application that is being monitored.
Client PID	The process ID of the client application that made the connection to the database.
OS User ID	The authorization ID used to access the operating system.
Application	Name of the application executable.
Status	The current status of the application.
Coord Node	In a multi-node system, this is the node number of the node where the selected application connected or attached to the instance.
Total Sort Time (ms)	The total elapsed time (in milliseconds) for all sorts that have been executed.
Total Sorts	The total number of sorts that have been executed.
Sort Overflows	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
Hash Joins	The total number of hash joins executed.
Hash Join Loops	The total number of times that a single partition of a hash join was larger than the available sort heap space.
Hash Join Overflows	The number of times that hash join data exceeded the available sort heap space.
Hash Join Small Overflows	The number of times that hash join data exceeded the available sort heap space by less than 10%.

I/O Tab

Column	Description
Agent ID	This is a system-wide unique identifier for the application.
Auth ID	The authorization ID of the user who invoked the application that is being monitored.
Client PID	The process ID of the client application that made the connection to the database.
OS User ID	The authorization ID used to access the operating system.

Column	Description
Application	Name of the application executable.
Status	The current status of the application.
Coord Node	In a multi-node system, this is the node number of the node where the selected application connected or attached to the instance.
Total I/O Time (ms)	The total time spent by application in performing buffered and non-buffered reads and writes.
Buffered I/O Time (ms)	The total time spent by application in performing buffered reads and writes.
Direct I/O Time (ms)	The total time spent by application in performing non-buffered reads and writes.
Total Reads	The number of reads issued against the database.
Total Writes	The number of writes issued against the database.
Logical Reads	The total number of db block gets and consistent gets (data read from memory) since the last refresh.
Physical Reads	The total number of physical reads performed on all datafiles since the last refresh.
Direct Reads	This is the number of read operations that do not use the buffer pool.
Direct Writes	This is the number of write operations that do not use the buffer pool.
Buffered Writes	This indicates the number of times a buffer pool data page was physically written to disk.
Pre-Fetch Wait Time (ms)	This is the time an application spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool.
Pre-Fetch Unread Pgs	This indicates the number of pages that the prefetcher read in that were never used.

Memory Tab

Column	Description
Agent ID	This is a system-wide unique identifier for the application.
Auth ID	The authorization ID of the user who invoked the application that is being monitored.
Client PID	The process ID of the client application that made the connection to the database.
OS User ID	The authorization ID used to access the operating system.
Application	Name of the application executable.
Status	The current status of the application.
Coord Node	In a multi-node system, this is the node number of the node where the selected application connected or attached to the instance.
Memory Used (KB)	Total memory pool usage for the application.
Priv Work HWM (KB)	This is the largest size (in kilobytes) reached by the private workspace.
Shr Work HWM (KB)	This is the largest size (in kilobytes) reached by shared workspaces.
Cache Overflows	This is the number of times that the catalog cache overflowed the bounds of its allocated memory.
Cache Lookups	This is the number of times that the catalog cache was referenced to obtain table descriptor information or authorization information.
Cache Inserts	This is the number of times that the system tried to insert table descriptor or authorization information into the catalog cache.

Column	Description
Catlg Cache Hit Ratio	The catalog cache hit ratio indicates how well the catalog cache is helping to avoid actual accesses to the catalog on disk. A high ratio indicates successful avoidance of actual disk I/O accesses.
Pkg Cache Hit Ratio	The package cache hit ratio indicates how well the package cache is helping to avoid reloading packages and sections for static SQL from the system catalogs as well as helping to avoid recompiling dynamic SQL statements. A high ratio indicates successful avoidance of these activities.
Priv Work Hit Ratio	The private workspace hit ratio indicates how well the private SQL workspace is helping to avoid having to initialize sections for SQL statements that are about to be executed. A high ratio indicate successful avoidance of this action.
Shr Work Hit Ratio	The shared workspace hit ratio indicates how well the shared SQL workspace is helping to avoid having to initialize sections for SQL statements that are about to be executed. A high ratio indicate successful avoidance of this action.
Appl Sect Hit Ratio	The application workspace hit ratio indicates how well the application SQL workspace is helping to avoid having to initialize sections for SQL statements that are about to be executed. A high ratio indicate successful avoidance of this action.

SQL Activity Tab

Column	Description
Agent ID	This is a system-wide unique identifier for the application.
Auth ID	The authorization ID of the user who invoked the application that is being monitored.
Client PID	The process ID of the client application that made the connection to the database.
OS User ID	The authorization ID used to access the operating system.
Application	Name of the application executable.
Status	The current status of the application.
Coord Node	In a multi-node system, this is the node number of the node where the selected application connected or attached to the instance.
Rows Read	This is the number of rows read from the table.
Rows Written	This is the number of rows changed (inserted, deleted, or updated) in the table.
Rows Read Per Tran	The average number of rows read per transaction.
Rows Written Per Tran	The average number of rows read per transaction.
Transactions	The total number of active transactions in that have not yet been committed, or are waiting on a blocking lock to complete.
Commits	This is the total number of commits initiated by the database manager.
Rollbacks	This is the total number of rollbacks initiated by the database manager.
Rows Selected	This is the number of rows that have been selected and returned to the application.
Rows Inserted	This is the number of row insertions attempted.
Rows Updated	This is the number of row updates attempted.
Rows Deleted	This is the number of row deletions attempted.
Select Stmt	This is the percentage of all executed statements that were SQL Select Statements.

Column	Description
DDL Stmt%	This is the percentage of all executed statements that were SQL DDL Statements.
Failed Stmt%	This is the percentage of all executed statements that were SQL Failed Statements.

Instance Page Statistics

Summary Tab

- Memory
 - [Sort Heap Utilization](#)
 - [Monitor Heap Utilization](#)
 - [Sort Heap \(MB\)](#)
 - [Private Memory \(MB\)](#)
- Sorts/Joins
 - [Piped Sorts Requested](#)
 - [Piped Sorts Rejected](#)
 - [Post Threshold Sorts](#)
 - [Post Threshold Hash Joins](#)
- FCM
 - [Buffer Utilization](#)
 - [Request Block Utilization](#)
 - [Message Anchor Utilization](#)
 - [Connection Entry Utilization](#)
- [Instance Identification](#)
 - Product Name
 - Service Level
 - Instance Type
 - Instance Name
 - Last Reset Time
 - Active Databases
 - Number of Nodes
 - Operating System

- [Monitor Switches](#)
 - Lock
 - Sort
 - Table
 - Bufferpool
 - Statement
 - Timestamp
 - Unit of Work
- Agent Statistics
 - [Idle](#)
 - [Stolen](#)
 - [Requests](#)
 - [Registered](#)
 - [Request Overflows](#)
 - [Waiting for Tokens](#)
 - [Assigned from Pool](#)
 - [Created Empty Pool](#)

Configuration Tab

- [Database Configuration](#)

Memory Pool Tab

- [Memory Pools](#)
- [Memory Pool Utilization](#)
- [Memory Pool Size](#)

Utilities Tab

- [Instance Utilities](#)

FCM Tab

- [FCM Throughput](#)
- [FCM Resource Utilization](#)
- [Buffer Utilization](#)
- [Message Anchor Utilization](#)
- [Connection Entry Utilization](#)
- [Request Block Utilization](#)

Other

- [Agent Utilization](#)
- [Percent of Agents Stolen](#)

Sort Heap Utilization

The amount of memory allocated for each sort may be some or all of the available sort heap size. Sort heap size is the amount of memory available for each sort as defined in the database configuration parameter *sortheap*. Shared sort memory utilization gives the percentage of the sort heap being used.

It is possible for a single application to have concurrent sorts active. For example, in some cases a SELECT statement with a subquery can cause concurrent sorts.

Metrics

Memory estimates do not usually include sort heap space. If excessive sorting is occurring, the extra memory used for the sort heap should be added to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Typically the shared sort utilization should be less than or equal to 70%. You should consider tuning the database if you see a utilization value greater than this.

Troubleshooting

To bring the sort heap utilization to an acceptable level, use the following guidelines:

- Examine the queries you are running on the database to see if you can add indexes to columns frequently accessed in the WHERE clause. This minimizes the use of the sort heap.
- If you are using dynamic bitmaps or hash join buffers in your queries, or your queries frequently require large sorts, increase the *sortheap* size.
- If you adjust the *sortheap* size, also look at the *sheaphres* database manager configuration parameter to see if it too needs to be adjusted

If you are adjusting the *sortheap* size, you may also benefit from rebinding your packages since the optimizer takes this parameter into account when determining the access paths.

Monitor Heap Utilization

The Monitor Heap Utilization statistic measures the consumption of monitor heap memory based on the memory pool. The utilization is calculated from using the following equation $(db2.pool_cur_size / db2.pool_max_size) * 100$.

Metric

The rate of utilization is measured by looking at the current memory heap pool size being used in relation to the maximum memory heap pool size. When the percentage reaches the maximum 100%, monitor operations may fail.

Troubleshooting

If you are experiencing trouble, consider increasing the monitor heap memory size.

Sort Heap (MB)

The Sort Heap metric is the total number of allocated pages of sort heap space for all sorts at the database manager level and at the moment the snapshot was captured.

Metric

Sort heap size is the amount of memory available for each sort as defined in the *sortheap* database configuration parameter. At the database manager level, it is the total sort heap space allocated in all active databases in the database manager.

Troubleshooting

If excessive sorting is occurring, you can add the extra memory used for the sort heap to the base memory requirements. Often, the larger the sort heap, the more efficient the sort. Use the information returned at the database manager level to help you adjust the *sheapthres* parameter.

Private Memory (MB)

This is the amount of private memory that the instance of the database manager has allocated at the time the snapshot was taken.

To change the amount of private memory allocated, change the minimum committed private memory configuration parameter (*min_priv_mem*). The default value is recommended, but you can use it to commit more memory to the database server. If you set the allocation too high, you can have a negative impact on the performance of non-DB2 applications.

Piped Sorts Requested

This metric gives the number of piped sorts that have been requested.

Metrics

Each active sort on the system allocates memory, which may result in sorting taking up too much of the available system memory. A piped sort is not accepted if the sort heap threshold is exceeded when the sort heap is allocated for the sort.

The sort list heap (*sortheap*) and sort heap threshold (*sheapthres*) configuration parameters help control the amount of memory used for sort operations. These parameters are also used to determine whether a sort will be piped.

Since piped sorts may reduce disk I/O, allowing more piped sorts can improve the performance of sort operations and possibly the performance of the overall system.

Troubleshooting

If piped sorts are being rejected, you might consider decreasing your sort heap or increasing your sort heap threshold. You should be aware of the possible implications of either of these options. If you increase the sort heap threshold there is the possibility that more memory will remain allocated for sorting. This could cause paging memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort.

Piped Sorts Rejected

The piped sorts rejects statistic is the number of piped sorts that were rejected by the database manager.

Metrics

Each active sort on the system allocates memory, which may result in sorting taking up too much of the available system memory. A piped sort is not accepted if the sort heap threshold is exceeded when the sort heap is allocated for the sort.

The sort list heap (*sortheap*) and sort heap threshold (*sheapthres*) configuration parameters help control the amount of memory used for sort operations. These parameters are also used to determine whether a sort will be piped.

Since piped sorts may reduce disk I/O, allowing more piped sorts can improve the performance of sort operations and possibly the performance of the overall system.

Troubleshooting

If piped sorts are being rejected, you might consider decreasing your sort heap or increasing your sort heap threshold. You should be aware of the possible implications of either of these options. If you increase the sort heap threshold there is the possibility that more memory will remain allocated for sorting. This could cause paging memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort.

Post Threshold Sorts

The post threshold sorts is the number of sorts that have requested heaps after the sort heap threshold has been exceeded.

Metrics

Under normal conditions, the Database Manager allocates sort heap using the value specified by the *sortheap* configuration parameter. If the amount of memory allocated to sort heaps exceeds the sort heap threshold (*sheapthres* configuration parameter), the database manager allocates sort heap using a value less than that specified by the *sortheap* configuration parameter.

Each active sort on the system allocates memory, which may result in sorting taking up too much of the system memory available. Sorts that start after the sort heap threshold has been reached may not receive an optimum amount of memory to execute. As a result, however, the entire system may benefit.

Troubleshooting

By modifying the sort heap threshold and sort heap size configuration parameters, the performance of sort operations and/or the overall system can be improved. If this element's value is high, you can:

- Increase the sort heap threshold (*sheapthres*) or,
- Adjust applications to use fewer or smaller sorts via SQL query changes.

Post Threshold Hash Joins

The post threshold hash joins statistic is the number of times that a hash join heap request was limited because of concurrent use of shared or private sort heap space.

Metrics

Each active sort on the system allocates memory, which may result in sorting taking up too much of the available system memory. A piped sort is not accepted if the sort heap threshold is exceeded when the sort heap is allocated for the sort.

The sort list heap (*sortheap*) and sort heap threshold (*sheapthres*) configuration parameters help control the amount of memory used for sort operations. These parameters are also used to determine whether a sort will be piped.

Since piped sorts may reduce disk I/O, allowing more piped sorts can improve the performance of sort operations and possibly the performance of the overall system.

Troubleshooting

If this value is large (greater than 5% of hash join overflows), the sort heap threshold should be increased.

Buffer Utilization

This element indicates the percentage of all FCM buffers that are currently being used by the fast communication manager.

Metrics

None.

Troubleshooting

You can use this information to tune *fcm_num_anchors*. If the utilization percentage is high, you should increase the *fcm_num_anchors* to ensure that operations do not run out of FCM message anchors. If the utilization is low, you can decrease *fcm_num_anchors* to conserve system resources.

Request Block Utilization

This element indicates the percentage of all request blocks that are currently being used by the fast communication manager.

NOTE: This metric is only applicable to DB2 version 7. In DB2 version 8, the maximum request blocks are adjusted dynamically and automatically

Troubleshooting

You can use this information to tune *fcm_num_rqb*. If the utilization percentage is high, you should increase the *fcm_num_rqb* to ensure that operations do not run out of FCM request blocks. If the utilization is low, you can decrease *fcm_num_rqb* to conserve system resources.

Message Anchor Utilization

This element indicates the percentage of all message anchors that are currently being used by the fast communication manager.

NOTE: This metric is only applicable to DB2 version 7. In DB2 version 8, the maximum message anchors are adjusted dynamically and automatically

Metrics

None.

Troubleshooting

You can use this information to tune *fcm_num_buffers*. If the utilization percentage is high, you should increase the *fcm_num_buffers* to ensure that operations do not run out of FCM buffers. If the utilization is low, you can decrease *fcm_num_buffers* to conserve system resources.

Connection Entry Utilization

The connection entry utilization element indicates the percentage of all connection entries that are currently being used by the fast communication manager.

NOTE: This metric is only applicable to DB2 version 7. In DB2 version 8, the maximum connection entries are adjusted dynamically and automatically.

Troubleshooting

You can use this information to tune *fc_num_connect*. If the utilization percentage is high, you should increase the *fc_num_connect* to ensure that operations do not run out of FCM connection entries. If the utilization is low, you can decrease *fc_num_connect* to conserve system resources.

Instance Identification

The following information about the database instance is displayed.

Database Instance Information	Description
Product Name:	The name of the database software.
Service Level:	The FixPack level.
Instance Type:	The type of DB2 server.
Instance Name:	The name of the DB2 instance.
DB2 Start Time:	The date and time that the database manager was started using the <code>db2start</code> command.
Active Databases:	The number of active databases running on the server.
Number of Nodes:	The number of nodes running on the instance.
Operating System:	The operating system the database is running on.

Monitor Switches

Monitor switches control the collection of potentially expensive data by the database manager. Each switch can be set to ON or OFF.

Lock

The lock monitor switch controls the collection of data related to the lock wait times and deadlocks.

Sort

The sort monitor switch controls the collection of data related to the number of heaps used and sort performance.

Table

The table monitor switch controls the collection of data related to the measure of activity (rows read/written).

Bufferpool

The bufferpool monitor switch controls the collection of data related to the number of reads and writes and the amount of time taken.

Statement

The statement monitor switch controls the collection of data related to the start/stop time and statement identification.

Timestamp

The timestamp monitor switch controls the collection of timestamps. This switch is ON by default.

Unit of Work

The unit of work monitor switch controls the collection of data related to start and end times and completion status.

Idle

An idle agent is one type of worker agent, that is an agent that carries out application requests but that has no fixed attachment to any given application. More specifically, an idle agent doesn't have a local database or outbound connection.

Metrics

The idle agents metric is the number of agents in the agent pool that are currently unassigned to an application and are, therefore, "idle."

Having idle agents available to service requests for agents can improve performance.

Troubleshooting

You can use this element to help set the *num_poolagents* configuration parameter.

Stolen

This is the number of times agents are stolen. Agents are stolen when an idle agent is reassigned from one application to another.

Metric

Agents stolen can be used along with *associate_agents_top* to evaluate the load the application places on the system.

Troubleshooting

If *agents_stolen* is high, try increasing the *num_poolagents* configuration parameter.

Requests

This metric is the number of requests for agents from the agent pool. As requests are made, idle agents are deployed.

Troubleshooting

Adjust the *num_poolagents* configuration parameter if there are there are more requests than agents available.

Registered

The agents registered metric is the number of agents registered in the Database Manager instance that is being monitored (coordinator agents and subagents).

Metrics

None.

Troubleshooting

You can use this element to help evaluate your setting for the *maxagents* configuration parameter.

Request Overflows

Request Overflows is the maximum number of agents associated with an application. The *max_agent_overflows* metric is the number of requests received after the *maxagents* configuration parameter has been reached.

Waiting for Tokens

The agents waiting for tokens statistic is the number of agents waiting for a token so they can execute a transaction in the database manager.

Metrics

Each application has a dedicated coordinator agent to process database requests within the Database Manager. Furthermore, each agent has to get a token before it can execute a transaction. The maximum number of agents that can execute Database Manager transactions is limited by the configuration parameter *maxcagents*.

Troubleshooting

You can use this element to help evaluate your setting for the *maxcagents* configuration parameter.

Assigned from Pool

This is the number of agents that have been assigned from the pool of agents available.

Troubleshooting

Use the *agents_created_empty_pool* in conjunction with the *agents_from_pool* metric to determine how often an agent has to be created because the pool is empty. If you see the ratio (Agents Created Due to Empty Agent Pool / Agents Assigned from Pool) is high, consider increasing the *num_poolagents* configuration parameter.

A low ratio suggests *num_poolagents* is set too high and rarely used agents are wasting resources. A high ratio may indicate the workload is too high for that node.

Created Empty Pool

This is the number of agents created because the agent pool ran dry. The metric includes the number of agents at startup (*num_intiagents*).

Troubleshooting

Use the *agents_created_empty_pool* in conjunction with the *agents_from_pool* metric to determine how often an agent has to be created because the pool is empty. If you see the ratio (Agents Created Due to Empty Agent Pool / Agents Assigned from Pool) is high, consider increasing the *num_poolagents* configuration parameter.

A low ratio suggests *num_poolagents* is set too high and rarely used agents are wasting resources. A high ratio may indicate the workload is too high for that node.

Database Configuration

The following information about the Database Manager (Instance) is available on the configuration tab:

Database Manager Information	Description
Parameter:	The parameter name.
Description:	The description of the parameter.
Value:	The value set for the parameter.
Modifiable:	Whether the parameter is modifiable or not.

Memory Pools

This section shows the memory pool usage details of the selected memory pool over time.

Usage Details	Description
Node:	Indicates the database partition number.
Memory Pool:	This is the kind of memory pool. Each pool type is only shown once.
Utilization:	The percentage of utilization of the selected memory pool.
High Watermark (MB):	The largest size of a memory pool since its creation.
Current Size (MB):	The current size of a memory pool.
Max Size (MB):	The internally configured size of a memory pool in DB2 UDB.

Metrics

The nature of *memory_pool* data elements varies between platforms. On Windows systems, no memory usage is reported at the database level, while on UNIX systems, memory is reported at the database level. Instead of reporting this memory at the database level, the system monitor for Windows systems reports it in instance -level snapshots. This divergence in reporting is due to differences in the underlying memory architecture between Windows systems and UNIX systems.

Memory Pool Utilization

This section shows the percentage of utilization of the selected memory pool.

Metrics

You can use this metric to see if a memory pool is nearly full. You can diagnose the problems with specific memory pools by monitoring their utilization over time.

Troubleshooting

If the value of the pool utilization is consistently close to or exceeds 100%, you may need to increase the configuration parameter associated with that pool.

Memory Pool Size

This section shows the current memory pool size for the selected application (or database level).

Instance Utilities

The Utilities tab of the Instance Page gives information on the utilities currently executing in the database manager (available in DB2 version 8 only). The Utility details are as follows:

Utility Details	Description
Node:	Indicates the database partition number.
Database:	Database on which the utility is operating.
ID:	The unique identifier corresponding to the utility invocation.
Type:	Class of the utility (Rebalance, Backup, Restore, Reorg, etc.).
Description:	A brief description of the work the utility is performing.
Start Time:	The time the utility was invoked.
% Complete:	Shows the completion status of the utility.

FCM Throughput

This section gives a detailed view of inter-node communication in a multipartition environment. The information shown here includes the connection status and buffers sent and received between various combinations of partitions in a multi-partitioned environment. The columns presented here include:

Source Node: Partition that sends the information.

Target Node: Partition that received the information.

Buffers Sent: Number of Buffers sent from source node to the target node.

Buffers Received: Number of Buffers sent from source node to the target node.

Connection Status: Status of the connection between nodes.

Metrics

If the total number of FCM buffers sent or received between selected nodes is high, you may want to redistribute the database, or move tables to reduce the inter-node traffic.

FCM Resource Utilization

The FCM Resource Utilization chart shows the following:

Node: Indicates the database partition number.

FCM Buffer Utilization: The current utilization and low watermark of FCM buffers.

FCM Message Anchor Utilization: The current utilization and low watermark of FCM message anchors.

FCM Connection Entries Utilization: The current utilization and low watermark of FCM connection entries.

FCM Request Block Utilization: The current utilization and low watermark of FCM request blocks.

Agent Utilization

Agent Utilization is the percentage of available database manager agents (maxagents) that are currently registered on the database manager (agents_registered).

Percent of Agents Stolen

Percent of Agents Stolen is the percentage of all agent requests—including the number of agents assigned from the agent pool (agents_from_pool), the number of agents created because the agent pool was empty (agents_created_empty_pool), and the number of agents stolen (agents_stolen)—that were agents stolen.

Other Views and Statistics

In addition to the Home view, Enterprise view, and the performance category views, Performance Center offers many other views. The tables below lists, by database platform, the other views available in Performance Center:

View	Oracle	SQL Server	Sybase	DB2
Alert Log	x	x	x	x
Archive	x			
Configuration Parameters		x	x	x
Health Index	x	x	x	x
Hot Objects	x			
Instance Parameters	x			
Lock	x	x	x	x
Operating System	x	x	x	x
Session Detail	x	x	x	x
SQL Server Logs		x		
Top Sessions	x	x	x	x
Top SQL	x	x		x
Trends	x	x	x	x

Archive View

- [Metrics](#)

To allow for point-in-time recovery, Oracle writes copies of redo log information to disk. When a database is running in archive log mode, a DBA, with proper backup techniques in place, can recover nicely from a database error and roll forward to almost any point in time as long as the proper archive logs are in place.

The I/O needed to write these archive logs is handled by Oracle's ARCH process. The Archive view allows archive files written by the ARCH process to be viewed by user-specified time frames. The table below describes the information available on the Archive view:

Column	Description
Date/Time	The timestamp of the archive log (when the log was written).
Title	The actual archive log file name and path.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Numerous archive files can be written to disk if there is heavy redo log activity. Batch jobs have the potential to move very fast; sometimes so fast that the online redo logs wrap back around before they have a chance to be archived. Messages indicating this has happened show up in the Oracle alert log. If this happens frequently, you should think about increasing the size of the online redo log files, or increasing the number of redo logs in general.

Seeing archive files written at a rate of more than one every 30-60 minutes can indicate the redo size is too small (or there is an above-average data modification load).

If you do not want to lose an archive file that can be needed for recovery and you are using Oracle 8 or later, you can take advantage of the feature where you can write archive files to more than one destination on disk. This feature also allows multiple ARCH processes to be invoked. Investigate using the Init.ora parameters `log_archive_dest_n` and `log_archive_max_processes`.

Always remember one thing with respect to archive files and running Oracle in archive log mode: Running out of archive file space on the server can halt all activity in a database. Make sure you have ample free space available on your archive drives. And, you should also implement a purge procedure for older archives in conjunction with your backup routine.

Health Index View

- [Metrics](#)

Performance Center's global and category-specific health indexes are fast and efficient indicators you can use to determine if a database is experiencing problems. They also locate the most problematic area(s). With the Health Index view, you can scan individual category indexes simultaneously and see, over time, where the problem areas reside.

Metrics

Generically speaking, you should investigate any index that falls below 90. Temporary dips in a health index graph should not be a cause for concern unless the dips form a pattern and occur on a predictable and continuous basis.

Hot Objects

The following tabbed pages are available on the Hot Objects view:

- [Hot Tables](#)
- [Hot Code](#)

NOTE: The Hot Objects view is for Oracle datasources.

Hot Tables

- [Metrics](#)

Certain objects in an Oracle database are accessed more than others. These objects can become a source of contention given certain conditions. The Hot Tables tab of the Hot Objects view identifies tables that are being frequently accessed through various SQL statements. The table below describes the information that Performance Center displays on the Hot Tables tab of the Hot Objects view:

Column	Description
Table Owner	The owner of the table.
Table Name	The name of the table.
Command Issued	The SQL statement command issued against the table.
Executions	The number of SQL executions the object has experienced.

Column	Description
Disk Reads	The number of estimated disk reads from the object.
Buffer Gets	The number of estimated logical I/O's from the object.
Rows Processed	The number of estimated rows processed from the object.

Metrics

DML activity against tables can cause contention for space management objects like free lists. Oracle9i and above provides automatic segment management, which can remove problems with free lists and the like.

Hot Code

- [Metrics](#)

Certain objects in an Oracle database are accessed more than others. Data objects can become a source of contention given certain conditions, while code objects rarely cause contention issues. The Hot Code tab of the Hot Objects view identifies code objects (procedure, functions, etc.) that are being frequently accessed through various SQL statements. The table below describes the information that Performance Center displays on the Hot Code tab of the Hot Objects view:

Column	Description
Object Owner	The owner of the object.
Object Name	The name of the object.
Object Type	The type of object (package, etc.)
# of Executions	The number of estimated executions for the object.
Loads	The number of times the object was loaded into the shared pool.
Locks	The number of locks the object has experienced.
Pins	The number of times the object was pinned in the shared pool.

Metrics

Often referenced code objects should be pinned in the shared pool using the Oracle DBMS_SHARED_POOL package. Objects with many executions and loads should be considered candidates for pinning.

Lock View

The following tabbed pages are available on the Lock view:

- [All Locks](#)
- [All User Locks \(Oracle only\)](#)
- [Blocking Locks](#)
- [Locks View for DB2](#)

All Locks Tab

The information on the All Locks tab of the Locks view depends on the target DBMS:

- [Oracle](#)
- [SQL Server](#)
- [Sybase](#)

All Locks Tab for Oracle

- [Metrics](#)

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, Oracle issues lock requests to carry out its internal duties. The All Locks tab of the Locks view displays information about all locks currently on a system. The table below describes the information available on the All Locks tab of the Locks view for Oracle:

Column	Description
SID	The session identifier of the session holding the lock.
User Name	The user account of the session holding the lock. NULL if it is a background process.
Lock Mode	The lock mode (EXCLUSIVE, SHARE, etc.)
Request Type	The type of lock requested by the session.
Object Name	The name of the object being locked.
Object Type	The type of object being locked (TABLE, etc.)
Lock Type	The type of lock (TRANSACTION, DML, etc.)
Lock ID 1	The lock identifier #1 (depends on type).
Lock ID 2	The lock identifier #2 (depends on type).

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Locks held for unusually long periods are candidates for further investigation. The application logic can be inefficient or the program may not be issuing COMMIT frequently enough.

All Locks Tab for SQL Server

- [Metrics](#)

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, SQL Server issues lock requests to carry out its internal duties. Performance Center displays information about all locks currently on a system on the All Locks tab of the Locks view and the Lock Detail grid on the Lock tab of the Contention Detail view.

The table below describes the information available on the tab and the grid:

Column	Description
SPID	The process ID of the process holding the lock.
Login	The login name of the process.
NT User	The operating system name of the process.
Database	The database where the locks are occurring.
Table Name	The name of the table involved in a lock. This will be NULL for non-table locks or table locks that take place in the tempdb database.
Ndx ID	The index ID involved in the lock.
Lock Type	The type of lock (database, table, row ID, etc.)
Lock Mode	The mode of the lock (shared, exclusive, etc.)
Lock Status	The status of the lock (waiting or granted).
Owner Type	Whether the lock came from a regular session or a transaction.
Program	The executable the process is using against the server.
BLK SPID	If nonzero, the process ID of the process blocking the requested lock. A value of zero indicates that the process is not blocked.
Wait Time	The time the process has waited for the lock, in milliseconds.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Command	The command the process is currently issuing.
NT Domain	The name of the Windows 2000/NT domain.

NOTE: The information in the Lock Detail grid is available in the [Lock Detail grid](#) on the Locks tab of the Contention Detail view and the All Locks tab of the Locks view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Locks held for unusually long periods are candidates for further investigation. The application logic may be inefficient or the program may not be issuing COMMIT frequently enough.

All Locks Tab for Sybase

- [Metrics](#)

To modify database information or structures, a user session must obtain a lock on the object to perform its task. In addition to user locks, Sybase issues lock requests to carry out its internal duties. The All Locks tab of the Locks view displays information about all locks currently on a system. The table below describes the information available on the All Locks tab of the Locks view for Sybase:

Column	Description
PID	The process ID.
User Name	The user name assigned to the process.
Database	The database in which the process is running.
Lock Type	The type of lock (database, table, row ID, etc.)
Object Name	The name of the object being locked.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Lock Page	The page number where the lock is currently applied.
Lock Class	The name of the cursor the lock is associated with (if any).
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Command	The command the process is currently issuing.
CPU Time	The CPU time accumulated for the current command.
Physical I/O	The current cumulative number of reads and writes issued by the process.
Mem Usage	The memory accumulated for the current command.
FID	The process ID of the worker process' parent.
Transaction	The name of any transaction.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Locks held for unusually long periods are candidates for further investigation. The application logic may be inefficient or the program may not be issuing COMMIT frequently enough.

All User Locks Tab

- [Metrics](#)

To modify database information or structures, a user session must obtain a lock on the object to perform its task. The All User Locks tab of the Locks view displays information about all user locks currently on a system. The table below describes the information available on the All User Locks tab of the Locks view:

Column	Description
User Name	The user account that holds the lock.
Terminal	The machine name of the client session.
SID	The unique Oracle identifier for the session.
Serial #	The serial number of the lock.
Table	The name of the object being locked.
Lock Mode	The lock mode (EXCLUSIVE, SHARE, etc.)
Request	The type of lock requested by the session.

NOTE: The All User Locks tab of the Locks view is only available for Oracle datasources.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Locks held for unusually long periods are candidates for further investigation. The application logic may be inefficient or the program may not be issuing COMMITs frequently enough.

Blocking Locks Tab

The information on the Blocking Locks tab of the Locks view depends on the target DBMS.

- [Oracle](#)
- [SQL Server](#)
- [Sybase](#)

Blocking Locks Tab for Oracle

- [Metrics](#)

Blocking-lock situations can make the database appear frozen, rivaling only a stuck archive in effect. A single blocking user has the potential to stop work for nearly all other processes on a small system, or can cause major headaches on large systems. Although Oracle supports unlimited row-level locking, blocking-lock situations do occur - sometimes frequently.

The Blocking Locks tab of the Locks view contains information relating to user accounts that are currently blocked and the sessions that are blocking them. The table below describes the information available on the Blocking Locks tab of the Locks view for Oracle:

Column	Description
Blocked SID	The session ID of the session waiting for the lock.
Blocked User	The user account of the session waiting for the lock.
Wait Time (sec)	The current wait time for the session, in seconds.
Blocking SID	The session ID of the session holding the offending lock.
Blocking User	The user account of the session holding the offending lock.
Lock Type	The type of lock (TRANSACTION, DML, etc.)
Lock Mode	The lock mode (EXCLUSIVE, SHARE, etc.)
Request Type	The type of lock being requested by the session.
Locked Object	The name of the object being locked.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

When a blocking lock is discovered, the DBA can quickly remedy the situation by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing. Other user processes then usually complete in an instant. Tools like Performance Center make it easier to discover the blocking-lock situation.

The culprit of blocking-lock scenarios is often the application design, or the SQL used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT...FOR UPDATE statements that place unnecessary restrictive locks on nearly every read operation. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Data warehouses, whose data is mostly read, can benefit from tablespaces set in read-only mode. Read-only mode signals to the other databases that exclusive locks need not be used for the data contained within the tablespace. This is especially helpful in Oracle Parallel Server environments and drastically reduces ping activity.

Blocking Locks Tab for SQL Server

- [Metrics](#)

Performance Center displays information about all blocking locks currently on a system on the Blocking Locks tab of the Locks view and the Blocking Lock Detail grid on the Blocking Lock tab of the Contention Detail view.

The table below describes the information available on the tab and the grid:

Column	Description
SPID	The process ID of the process holding the lock.
Login	The login name of the process.
NT User	The operating system name of the process.
Database	The database where the locks are occurring.
Table Name	The name of the table involved in a lock. This will be NULL for non-table locks or table locks that take place in the tempdb database.
Ndx ID	The index ID involved in the lock.
Lock Type	The type of lock (database, table, row ID, etc.).
Lock Mode	The mode of the lock (shared, exclusive, etc.).
Lock Status	The status of the lock (waiting or granted).
Owner Type	Whether the lock came from a regular session or a transaction.
Program	The executable the process is using against the server.
BLK SPID	If nonzero, the process ID of the process blocking the requested lock. A value of zero indicates that the process is not blocked.
Wait Time	The time the process has waited for the lock, in milliseconds.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Command	The command the process is currently issuing.
NT Domain	The name of the Windows 2000/NT domain.

NOTE: The information in the Blocking Lock Detail grid is available in the [Blocking Lock Detail grid](#) on the Blocking Locks tab of the Contention view and the Blocking Locks tab of the Locks view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

When a blocking lock is discovered, the DBA can quickly remedy the situation by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing. Other user processes then usually complete in an instant. Tools like Performance Center make it easier to discover the blocking-lock situation.

Blocking Locks Tab for Sybase

- [Metrics](#)

The Blocking Locks tab of the Locks view contains information relating to user accounts that are currently blocked and the sessions that are blocking them. The table below describes the information available on the Blocking Locks tab of the Locks view for Sybase:

Column	Description
Holding PID	The process ID that owns the blocking lock.
Holding User	The user account of the session holding the offending lock.
Waiting PID	The session PID of the session waiting for the lock.
Waiting User	The user account of the session waiting for the lock.
Database	The database in which the process is running.
Object Name	The table on which the lock is being held.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Lock Type	The type of lock being applied.
Time Blocked	The time that a process has been waiting for the lock, in seconds.
Lock Page	The page number where the lock is currently applied.
Lock Class	The name of the cursor the lock is associated with (if any).
Holding Host	The name of the host computer with the blocking lock.
Waiting Host	The name of the host computer waiting for the lock.
Holding Program	The program the process is running that has the lock.
Waiting Program	The program the process is running that is waiting for the lock.
Holding Command	The command being issued by the process holding the lock.
Waiting Command	The command being issued by the process waiting for the lock.
CPU Time	The CPU time accumulated for the current command.
Physical I/O	The physical I/O accumulated for the current command.
Mem Usage	The memory accumulated for the current command.
Holding FID	The process ID of the worker process' parent that has the lock.
Waiting FID	The process ID of the worker process' parent that is waiting for the lock.
Transaction	The name of the associated transaction (if any).

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

When a blocking lock is discovered, the DBA can quickly remedy the situation by issuing a KILL against the offending process. This eliminates the user's stranglehold on the objects the user was accessing. Other user processes then usually complete in an instant. Tools like Performance Center make it easier to discover the blocking-lock situation.

Locks View for DB2

The Locks view displays all processes that are currently holding locks on an IBM DB2 UDB database. The following sections of this view are available to display lock information:

- [Applications](#)
- [Locks Held Tab](#)

- [Locks Waiting Tab](#)
- [Unit of Work Tab](#)

Applications

This section lists the following lock information for all applications:

Agent ID – The application handle of the agent holding a lock for which this application is waiting.

Auth ID – The authorization ID of the user who invoked the application that is being monitored.

OS User ID – The authorization ID used to access the operating system.

Client PID – The process ID of the client application that made the connection to the database.

Application – Name of the application executable.

Status – The lock's status (waiting or granted).

Locks Held – The number of locks on the lock being held.

Locks Waiting – Indicates the number of agents waiting on a lock

Lock Wait Time (ms) – The current amount of wait time for the process, in milliseconds.

Timeouts – The number of times that a request to lock an object timed out without being granted.

Deadlocks – Processes that cannot proceed because they are waiting on a set of resources held by each other or held by other processes.

Locks Held Tab

This tab displays all the locks held by the selected application in the Applications list. The following data is available:

Lock Mode – The type of lock being held.

Object Type – The type of object against which the application holds a lock (for object-lock-level information), or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information).

Table Schema – Schema of the table that the lock is on.

Table Name – Name of the table that the lock is on. This element is only set if Object Type indicates Table.

Tablespace – The name of the table space against which the lock is held.

Lock Status – The lock's status (waiting or granted).

Escalation – Indicates whether a lock request was made as part of a lock escalation.

Locks Waiting Tab

This tab displays all the locks waiting by the selected application in the Applications list. The following data is available:

Agent ID – The application handle of the agent holding a lock for which this application is waiting.

Application ID – The application ID of the application that is holding a lock on the object that this application is waiting to obtain.

Lock Mode – The type of lock being held.

Mode Requested – The lock mode being requested by the application.

Object Type – The type of object against which the application holds a lock.

Table Schema – Schema of the table that the lock is on.

Table Name – Name of the table that the lock is on. This element is only set if Object Type indicates Table.

Tablespace – The name of the table space against which the lock is held.

Wait Start Time – The date and time that this application started waiting to obtain a lock on the object that is currently locked by another application

Escalation – Indicates whether a lock request was made as part of a lock escalation.

Unit of Work Tab

This tab displays the SQL statement text for the selected application. The statement is available if the selected application is in lock wait status or is the thread blocking other applications. This enables you to easily identify what SQL statements are causing lock wait conditions and to help diagnose deadlock scenarios.

Click **Explain SQL** to view an explain plan for the statement.

Operating System View

The Operating System view displays vital operating system statistics on the following tabbed pages:

- [Summary](#)
- [CPU](#)
- [Processes](#)
- [I/O](#)
- [Memory](#)
- [Space](#)
- [Network](#)

To use integrated security, or gather certain operating system statistics, there are two main things to know:

- 1 You must enable the Performance Center Server for operating system monitoring. To enable it, you must select the Enable operating system monitoring option on the [Machine tab](#) of the Datasource Properties dialog.
- 2 You must supply the credentials necessary to view these statistics. To be able to view performance counters on a remote computer, Microsoft requires specific permissions on the remote computer that you want to monitor.

Because the Performance Center Server collects data using the registry, monitoring a remote computer requires the use of the Remote Registry Service. If the service stops due to failure, the system restarts it automatically only once. Therefore, if the service stops again, you must manually restart it. You can change this default behavior by modifying the properties for Remote Registry Service. To access service properties, see Services under Services and Applications in Computer Management or see Administrative Tools. You can also check the Event Viewer's Application and System Logs for events that might have stopped the service.

Remote data collection also requires access to specific registry subkeys and system files. To provide remote access to the registry to collect data on remote systems, Microsoft requires that users have a minimum of Read access to the Winreg subkey in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers. By default, members of the Administrators group have Full Control access and members of the Backup Operators group have Read access. Microsoft also requires that users have Read access to the registry subkey that stores counter names and descriptions used by System Monitor, HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\LanguageID, where *LanguageID* is the numeric code for the spoken language for the operating system installation. (For English, the subkey is Perflib\009.) By default, Microsoft gives Full Control access to the System account and members of the Administrators and Creator Owners groups. Therefore, a local user on a server who is not logged in as an administrator cannot see performance counters.

In addition, users might also require read access to the files that supply counter names and descriptions to the registry, Perf*.dat and Perfh*.dat. (where the asterisk is a wildcard character representing the specific language code; for English, these are Perf009.dat and Perfh009.dat.) If these files reside on an NTFS volume, to have access to them, the access control lists (ACLs) on these files must specify that the user has such access. By default, members of the Administrators and Interactive groups have sufficient access.

OS Page Statistics

In many scenarios, an optimally tuned database may not perform well because there are constraints imposed by the system where the database is running. These constraints may include processes competing with the database server for resources (CPU, I/O, or Memory), a slow CPU, insufficient or slow I/O devices, and insufficient memory. In the OS Statistics page of Performance Center you can examine operating system metrics for the following platforms:

- [AIX](#)
- [HP-UX](#)

NOTE: To view processor info and swap disk info on an HP-UX box, you need to login as ROOT in the OS login.

- [Linux](#)
- [Solaris](#)
- [Unix](#)
- [Windows XP, 2000, and NT](#)

Summary Tab

The OS Summary tab displays the following statistics to communicate the general overall performance levels of the operating system:

- [Disk Time](#)
- [Load Average](#)
- [Processor Time](#)
- [Paged Memory Used \(Windows\)](#)
- [Swap Memory Used \(AIX, HP-UX, Linux, Solaris, Unix\)](#)
- [Average Disk Queue](#)
- [Network Output Queue \(Windows\)](#)
- [Network Queue \(Solaris\)](#)
- [Page Faults/Sec](#)
- [Processor Queue](#)
- [Processor Speed](#)
- [Processor](#)
- [Available Paged Memory \(Windows\)](#)
- [Available Physical Memory](#)
- [Available Swap Memory \(AIX, HP-UX, Linux, Solaris, Unix\)](#)
- [Total Paged Memory \(Windows\)](#)
- [Total Physical Memory](#)
- [Total Swap Memory \(AIX, HP-UX, Linux, Solaris, Unix\)](#)
- [Free Disk Space](#)
- [Total Disk Space](#)
- [Used Disk Space](#)
- [Number of Logins](#)
- [Number of Processes](#)
- [Number of Processors](#)
- [Top CPU Process](#)
- [Top I/O Process](#)
- [Top Memory Process](#)

Processor Time

The Processor Time statistic indicates the percentage of time the processor is working. This counter is a primary indicator of processor activity.

Metrics

If your computer seems to be running sluggishly, this statistic could be displaying a high percentage.

Troubleshooting

Upgrade to a processor with a larger L2 cache, a faster processor, or install an additional processor.

Processor Speed

The Processor Speed statistic displays the speed of the active processor in MHz. The speed is approximate.

Processor

The Processor Statistic displays the type of processor currently in use, for example, GenuineIntel.

Disk Time

The Disk Time statistic is the percentage of elapsed time that the selected disk drive/device was busy servicing read or write requests.

Metrics

You should avoid consistently seeing values for this statistic greater than 90%.

Troubleshooting

Add more disk drives and partition the files among all of the drives.

Load Average

The Load Average statistic represents the system load averages over the last 1, 5, and 15 minutes.

Metrics

High load averages usually mean that the system is being used heavily and the response time is correspondingly slow.

Paged Memory Used

The Paged Memory Used statistic is the ratio of Commit Memory Bytes to the Commit Limit. Committed memory is where memory space has been reserved in the paging file if it needs to be written to disk. The commit limit is determined by the size of the paging file. As the paging file increases, so does the commit limit.

NOTE: This statistic is available for the Windows platform.

Metrics

This value displays the current percentage value only and not an average. If the percentage of paged memory used is above 90%, you may be running out of memory.

Troubleshooting

Increase the size of page file.

Number of Processors

This statistic displays the number of processors currently in use.

Swap Memory Used

The Swap Memory Used statistic is the percentage of swap space currently in use.

Metrics

If the percentage of swap memory used is above 90%, you may be running out of memory.

Troubleshooting

Increase the size of your swap files.

Average Disk Queue

The Average Disk Queue statistic is the average number of both read and write requests that were queued for the selected disk during the sample interval.

Metrics

This metric is useful in identifying I/O related bottlenecks. If the disk queue lengths for certain disks are consistently much higher than others, you may need to redistribute the load among available disks. If the disk queues lengths for all disks are consistently large, and you see a high amount of I/O activity, your disks may be inefficient.

Troubleshooting

Some things you can do if you have problems with this statistic include:

- Redistribute the data on the disk with the large average disk queue to other disks.
- Upgrade to faster disk(s).

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics

This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Processor Queue

The Processor Queue Length statistic is the number of threads in the processor queue.

Metrics

Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting

A sustained high value in the Processor Queue could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Network Output Queue/Network Queue

The Network Output Queue Length statistic is the number of threads in the processor queue.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics

Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

Troubleshooting

A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Available Physical Memory

The Available Physical Memory statistic represents the amount of RAM available to all processes.

Metrics

This counter displays the last observed value only and not an average. Use this value with the Total physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Troubleshooting

Some things you can do if you have problems with this statistic include:

- Check the running processes to see if there are any memory leaks.
- Stop any services that are not required.
- Install additional RAM.

Available Paged Memory

The Available Paged Memory statistic shows the amount of virtual memory available for the processes.

NOTE: This statistic is available for the Windows platform.

Metrics

If the available virtual memory is less than 10% of the total virtual memory, your system may run out of memory.

Troubleshooting

Increase the size of page file.

Available Swap Memory

The Available Swap Memory statistic represents the amount of virtual memory available for the processes.

Metrics

If the available Available Swap Memory is less than 10% of the total Swap Memory, your system may run out of memory.

Troubleshooting

Increase the size of swap files.

Total Physical Memory

The Total Physical Memory statistic shows the amount of physical memory installed on your computer.

Metrics

This is an informational metric and displays the total amount installed on the machine. Use this value with the available physical memory and paging metrics (Memory details page). If the available physical memory is very small compared to this value, and the paging activity is high, your system may be running low on memory.

Total Paged Memory/Total Swap Memory

The Total Paged Memory statistic shows the maximum amount of virtual memory available to all processes.

NOTE: The name of this statistic depends on the platform of the operating system.

Metrics

It is recommended that this value is between 1.5 to 3 times the amount of RAM on the system.

Used Disk Space

The Used Disk Space statistic shows the amount of allocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Total Disk Space

Total Disk Space displays the total allocated and unallocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations, here are two:

- 1 If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- 2 You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Free Disk Space

The Free Disk Space statistic shows the unallocated space, in megabytes on all logical disk drives.

Troubleshooting

There are many things a DBA can do to ensure that a database does not encounter a space problem due to physical space limitations:

- If a database currently resides on a disk that has little free space, you can add more files to the database. Of course, you should add the new files to other physical hard disks that can accommodate a growing database.
- You should examine hard disks with shrinking disk space to see if you can relocate or delete files to allow more free space.

Top Memory Process

Top Memory Process shows the current process that is consuming the most amount of memory. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics

If you are running out of memory on the system, this is a quick way to identify the top memory user. If the displayed process is using a significant portion of the total memory, it could be causing the memory issues.

Processes Overview

The Processes Overview of the OS Summary includes the following sections:

- [Top CPU Process](#)
- [Top I/O Process](#)
- [Top Memory Process](#)

Top CPU Process

Top CPU Process shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Metrics

If the amount of CPU time used by this process is close to 100% and the CPU usage is very high, this process may be the bottleneck on the server.

Troubleshooting

Investigate the process further to see if it is in an inconsistent state. Also, look at minimum requirements for CPU speed for the process. You may need to upgrade your CPU.

Top I/O Process

The Top I/O Process statistic shows the current process that is consuming the most amount of CPU. The information displayed is dependent on the platform of the operating system. Information displayed includes the name of the process, process ID, amount of memory consumed expressed in KB, amount of CPU expressed as a percentage, the amount of Major Page Faults, and the amount of I/O expressed in KB/sec.

Number of Logins

This statistic displays the total number of logins on the server.

Number of Processes

This statistic displays the total number of processes on the server.

CPU Tab

The CPU tab of the OS Detail includes the following sections:

- [Context Switches/Sec](#)
- [CPU Utilization](#)
- [Interrupts/Sec](#)
- [Processor Queue Length](#)

CPU Utilization

The CPU Utilization section includes the following information:

- [% Privileged Time](#)
- [% User Time](#)

% Privileged Time

The % Privileged Time statistic is the percentage of elapsed time that the process threads spent executing code in privileged mode.

NOTE: For Windows systems, when a Windows system service is called, the service will often run in privileged mode to gain access to system-private data. Such data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. These kernel commands, are considered privileged to keep the low-level commands executing and prevent a system freeze. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.

Metrics

The ideal range should be 0-40% (less than 40% indicates excessive system activity).

Troubleshooting

If your CPU consistently runs at less than 40% you may need to upgrade your system to include a faster processor(s).

% User Time

The % User Time statistic is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems. The alternative, privileged mode, is designed for operating system components and allows direct access to hardware and all memory. The operating system switches application threads to privileged mode to access operating system services. This counter displays the average busy time as a percentage of the sample time.

Metrics

If the Privileged Time is high in conjunction with Physical Disk Reads, consider upgrading the disk I/O subsystem.

Interrupts/Sec

The Interrupts/Sec statistic is the average rate, in incidents per second, at which the processor received and serviced hardware interrupts. It does not include deferred procedure calls (DPCs), which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended. The system clock typically interrupts the processor every ten milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Metrics

The ideal range should be 0-5000. A number greater than 5000 indicates possible excessive hardware interrupts; justification is dependent on device activity.

Context Switches/Sec

The Context Switches/Sec section shows the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service.

Metrics

The ideal range should be between 0-10,000. GA number greater than 10,000 may indicate too many threads contending for resources.

Processor Queue Length

The Processor Queue Length statistic is the number of threads in the processor queue. There is a single queue for processor time even on computers with multiple processors.

NOTE: For Windows systems, unlike the disk counters, this counter shows ready threads only, not threads that are running.

Metrics

A sustained high value in the Processor Queue Length could indicate that a processor bottleneck has developed due to threads of a process requiring more process cycles than are available. If this is the case, you should look at installing a faster (or an additional) processor.

Processes Tab

The Processes tab of the OS Detail page succinctly communicates the general overall performance levels of processes. The columns available in this table depend on the platform of operating system. The table below describes the information available in the table on this tab:

Column	Description
Process	The name of the process.
User	The user of the process.
ID	The ID Process is the unique identifier of this process. ID Process numbers are reused, so they only identify a process for the lifetime of that process.
CPU	The CPU is the percentage of elapsed time that all of process threads used the processor to execution instructions.
User Mode	The User Mode is the percentage of elapsed time that the process threads spent executing code in user mode.
Memory WINDOWS ONLY	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory (MB)	Memory is the current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Memory	Memory is the percentage of the memory used of the total memory.
Active Memory	Active Memory is the amount of committed virtual memory, in bytes for this process. Active memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.

Column	Description
I/O Data	The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network and device I/Os.
Elapsed Time	The total elapsed time, in seconds, that this process has been running.
Thread Count	The number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Handle Count	The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.
Priority	The current base priority of this process. Threads within a process can raise and lower their own base priority relative to the process' base priority.
Creating Proc ID	The Creating Process ID value is the Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Page Faults/Sec	Page Faults/Sec is the rate at which page faults by the threads executing in this process are occurring. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and hence already in main memory, or if it is in use by another process with whom the page is shared.
Page File	Page File is the current number of kilobytes that this process has used in the paging file(s). Paging files are used to store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory.
Private	Private is the current size, in kilobytes, of memory that this process has allocated that cannot be shared with other processes.

I/O Tab

The table below describes the information available in this section:

Column	Description
Disk	The disk number assignment.
Reading (KB/s)	The amount of bytes read from the device.
Writing (KB/s)	The amount of bytes written to the device.
Disk Read Time	Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.
Disk Write Time	Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.
Disk Time	Disk Time is the percentage of elapsed time that the selected disk was busy servicing requests.
Avg. Read Queue	Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.
Avg. Write Queue	Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.
Disk Reads/Sec	Disk Reads/Sec is the rate of read operations on the disk.
Disk Writes/Sec	Disk Writes/Sec is the rate of write operations on the disk.

Memory Tab

The Memory tab of the OS Detail page includes the following sections:

- [Cache Efficiency](#)
- [Free Physical](#)
- [Free Paged](#)
- [Paging Activity](#)
- [Page Faults](#)
- [Total Physical](#)
- [Total Paged](#)

Paging Activity

The Paging Activity section includes the following statistics:

- [Pages Input/Sec](#)
- [Pages Output/Sec](#)

Pages Input/Sec

The Pages Input/Sec statistic is the number of pages read from disk to resolve hard page faults. Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk.

Metrics

This value was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of pages, and can be compared to other counts of pages, such as Memory: Page Faults/sec, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Pages Output/Sec

The Pages Output/Sec statistic is the number of pages written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory. A high rate of pages output might indicate a memory shortage.

Metrics

Windows NT writes more pages back to disk to free up space when low in physical memory. This counter counts numbers of pages, and can be compared to other counts of pages, without conversion. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

Although it never hurts to have as much physical memory as your system can handle, there are some things you can check within your system to alleviate the memory bottleneck.

- Check to see if you have any drivers or protocols that are running but not being used. They use space in all memory pools even if they are idle.
- Check to see if you have additional space on your disk drive that you could use to expand the size of your page file. Normally, the bigger the initial size of your page file, the better, in performance terms.

Free Physical

The Free Physical statistic is the amount of physical memory that is uncommitted.

Metrics

None.

Free Paged

The Free Paged statistic is the amount of uncommitted virtual memory.

Metrics

None.

Total Physical

The Total Physical statistic is the total physical memory available.

Metrics

None.

Total Paged

The Total Paged statistic is the amount of total virtual memory, in bytes. Used Memory is the physical memory that has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive.

Metrics

None.

Page Faults/Sec

The Page Faults/Sec statistic is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set. This counter includes both hard faults and soft faults.

Metrics

This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

Troubleshooting

If the number of page faults remains consistently high, you can check with your Windows System Administrator for further investigation. Often, large numbers of page faults are not a problem so long as they are soft faults. However, hard faults, that require disk access, can cause delays.

Cache Efficiency

The Cache Efficiency section of the Memory tab succinctly communicates the general overall performance levels of the server's memory. The following statistics are available in this section:

- [Copy Read Hits%](#)
- [Data Map Hits%](#)
- [MDL Read Hits%](#)
- [Pin Read Hits%](#)

Copy Read Hits %

The Copy Read Hits % statistic is the percentage of cache copy read requests that hit the cache and does not require a disk read to provide access to the page in the cache.

Metrics

When the page is pinned in the memory, the page's physical address in the file system cache will not be altered. A copy read is a file read operation where a page in the cache is copied to the application's buffer. Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Data Map Hits %

The Data Map Hits % statistic is the percentage of data maps in the file system cache that could be resolved without having to retrieve a page from the disk.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

MDL Read Hits %

The MDL Read Hits % statistic is the percentage of Memory Descriptor List Read requests to the file system cache that hit the cache and does not require disk access to provide memory access to the pages in the cache.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Pin Read Hits %

The Pin Read Hits % statistic is the percentage of pin read requests that hit the file system cache and does not require a disk read in order to provide access to the page in the file system cache.

Metrics

Because this value reflects hits, it ideally should be close to 100%. An amount below 100% indicates misses.

Troubleshooting

Adding physical memory to a server results in a larger file system cache, which is generally more efficient. Defragmenting disks also helps, putting related pages in the cache together and thereby improving the cache hit rate.

Space Tab

The Space tab of the OS Detail page includes the following sections:

- [Disk Space Free](#)
- [Disk Space Detail](#)

Disk Space Free

The Disk Space Free metric displays the amount of free disk space in megabytes.

Metric

None.

Disk Space Detail

The Disk Space Detail section of the Space tab succinctly communicates the general overall performance levels of the server's disks and space allotment. The table below describes the statistics in this section:

Statistic	Description
Partition	The drive letter of the disk.
Total Space	Total size of the disk/device's capacity expressed in MBs.
Used Space	Amount of MBs currently allocated on the particular disk/device.
Free Space	Amount of MBs currently unallocated and free on the particular disk/device.

Network Tab

The Network tab of the OS Detail page succinctly communicates the general overall performance levels of the server's networking. The statistics available in this section depend on the platform of operating system. The table below describes the information available in this section:

Column	Description
Network Interface	The name of network interface.
INET Address/Address	The IP address assigned to the network interface.
Pkts Sent/Sec	The number of packets sent per second.
Pkts Received/Sec	The number of packets received per second.
Sent (KB/Sec)	The number of bytes sent per second.
Received (KB/Sec)	The number of bytes received per second.
Out Pkts Discarded	The number of outbound packets discarded.
In Pkts Discarded	The number of inbound packets discarded.
Out Pkt Errors	The number of outbound packet errors.
In Pkt Errors	The number of inbound packet errors.
Queue Length	The queue length.
Collisions	The number of collisions.
Packets Discarded	The number of packets discarded.

Session Detail View

The Session Detail view is available for the following database platforms:

- [Oracle Session Detail View](#)
- [SQL Server Session Detail View](#)
- [Sybase Session Detail View](#)
- [DB2 Session Detail View](#)

Oracle Session Detail View

The following tabbed pages are available on the Session Detail view for Oracle:

- [Session Contention](#)
- [Session I/O](#)
- [Session Memory](#)
- [Session Network](#)
- [Session Objects](#)
- [Session SQL](#)
- [Session Statistics](#)

Session Memory Tab for Oracle

- [Metrics](#)

The Session Memory tab of the Session Detail view presents the statistics surrounding a session's memory usage. The table below describes the information available on the Session Memory tab of the Session Detail view for Oracle:

Column	Description
Statistic	The name of the memory related statistic.
Value	The cumulative value for the memory statistic.
Cache Hit Ratio	The percentage of data obtained from memory access vs. physical I/O.

Metrics

Sessions with abnormally high memory usage can affect overall performance at the server level, as this memory (PGA and UGA) is allocated outside of the Oracle SGA (unless the multi-threaded server option is being used). If cache hit ratios at the session level are lower than 85 percent, data access can be inefficient.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session I/O Tab for Oracle

- [Metrics](#)

The Session I/O tab of the Session Detail view presents the statistics surrounding a session's I/O activity. The table below describes the information available on the Session I/O tab of the Session Detail view for Oracle:

Column	Description
Statistic	The name of the I/O related statistic.
Value	The cumulative value for the I/O statistic.

Metrics

Seeing high values for physical reads and writes can indicate an inefficient session. Large numbers of physical reads can imply a session with too many large table scans or inefficient SQL operations. Large numbers of physical writes can be okay for sessions inputting large volumes of data into the database. Or, they could also indicate a session involved in heavy disk sort activity.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session Contention Tab for Oracle

- [Metrics](#)

The Session Contention tab of the Session Detail view presents information relating to resources on which the current session is waiting. The [first grid](#) displays user waits. The [second grid](#) displays user locks. The table below describes the information available in the User Waits grid on the Session Contention tab of the Session Detail view for Oracle:

Column	Description
Wait Cause	The wait event being experienced by the session.
Program	The program the session is executing against Oracle.
Seconds	The number of seconds the session has spent in the wait.

Column	Description
State	The status of the wait event (WAITING, etc.)

The table below describes the information available in the User Locks grid on the Session I/O tab of the Session Detail view for Oracle:

Column	Description
User	The user account being used by the session.
Terminal	The client machine name used by the session.
SID	The unique Oracle identifier for the session.
Serial #	The serial number for the session.
Table	The object locked by the session.
Lock Mode	The lock mode used by the session.
Title	The lock request issued by the session.

Metrics

You can ignore some waits, like the SQL*Net more data from client and SQL*Net message from client. Others, like enqueue waits, can be indicative of a lock contention problem. Waits, like db file scattered reads, can indicate sessions involved in long table scan operations.

Locks that are held for unusually long periods require further investigation. The application logic can be inefficient or the program is not issuing COMMITs frequently enough. The culprit of blocking-lock scenarios is usually the application design, or the SQL used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the correct SQL to get the job done is an art. Most DBAs who have had to face Oracle Forms applications have suffered through the dreaded SELECT...FOR UPDATE statements. These place unnecessary restrictive locks on nearly every read operation. The key to avoiding lock contention is to process user transactions as quickly and efficiently as possible.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session Objects Tab for Oracle

- [Metrics](#)

The Session Objects tab of the Session Detail view presents information regarding the objects in use by a session. The table below describes the information available in the Objects Accessed grid on the Session Objects tab of the Session Detail view for Oracle:

Column	Description
Owner	The owner of the object.
Type	The type of object (TABLE, VIEW, etc.).
Object	The name of the object.

The Rollback Segments Accessed grid displays the names of rollback segments currently being used by the session.

Metrics

Once you have an idea of which objects your users access most often, you can refine some processes to facilitate access to them. You can use the Oracle 8 concept of the KEEP buffer cache to force Oracle to hold often-referenced data for data objects. The KEEP buffer cache is ideal for holding small look-up tables. If you have an earlier version of Oracle or do not want to split up your current buffer cache, you can use the CACHE table attribute to encourage Oracle to keep data blocks of CACHE'd tables at the most recently used end of the LRU buffer cache chain.

If you consistently see a session with active rollbacks, it can indicate locks are being held for long durations. It can also indicate that a session is using code without frequent enough COMMIT points.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session Network Tab for Oracle

- [Metrics](#)

The Session Network tab of the Session Detail view presents information about requests being sent to and from the database with respect to the current session. The table below describes the information available on the Session Network tab of the Session Detail view for Oracle:

Column	Description
Statistic	The name of the SQL*Net related statistic.
Value	The cumulative value of the SQL*Net related statistic.

Metrics

None.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session SQL Tab for Oracle

- [Metrics](#)

The Session SQL tab of the Session Detail view presents information relating to any current SQL issued by the session as well as any SQL previously issued by the session.

Metrics

To determine access paths, you should export and run through an EXPLAIN PLAN session any SQL that you suspect of inefficient access.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Session Statistics Tab for Oracle

- [Metrics](#)

The Session Statistics tab of the Session Detail view presents information relating to all recorded performance and miscellaneous statistics for the current session. The table below describes the information available on the Session Statistics tab of the Session Detail view for Oracle:

Column	Description
Statistic	The name of the session related statistic.

Column	Description
Value	The cumulative value of the session related statistic.

Metrics

None.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

SQL Server Session Detail View

The Session Detail view down provides a granular look at the details when a process is acting in a way that merits further investigation. It also presents data about a particular process in a way that makes it easier to understand and view apart from all other processes that SQL Server is running.

The following tabbed pages are available on the Session Detail view for SQL Server:

- [All Locks](#)
- [Blocked By](#)
- [Blocking](#)
- [Overview](#)
- [SQL](#)

Overview Tab for SQL Server

- [Metrics](#)

The Overview tab of the Session Detail view displays information to analyze the details of a particular process. The tables below describe the statistics for each category on the Overview tab of the Session Detail view for SQL Server. The available categories are:

- [Contention](#)
- [General](#)
- [I/O](#)
- [Memory](#)
- [Network](#)
- [Users](#)

General Statistics

The table below describes the statistics in the General category on the Overview tab of the Session Detail view:

Statistic	Description
SPID	The SQL Server process ID. Unique value across all processes.
Login Name	The SQL Server login name of the process.
NT User	If using Windows Authentication, the name of the Windows NT user for this process. If using a trusted connection, the Windows NT user name.

Statistic	Description
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Database	The database in which the process is running.
Program	The executable the process is using against the server.
Host	The machine name that originated the process.

Memory Statistics

The table below describes the statistics in the Memory category on the Overview tab of the Session Detail view:

Statistic	Description
Memory Usage	The number of pages in the procedure cache (SQL Cache) that are currently allocated to this process. A negative number indicates that pages are being released (freed) from this process.
Buffer Cache Hit Ratio	The percentage of data page requests by this process that is available in memory as opposed to performing a physical I/O to disk.

Contention Statistics

The table below describes the statistics in the Contention category on the Overview tab of the Session Detail view:

Statistic	Description
Blocked By	If the process is being blocked, the SPID of the blocking process. A value of zero means that the process is not blocked.
Wait Time	The number of milliseconds that the process has been waiting to be serviced. A value of zero indicates that the process is not waiting.
Last Wait Type	The last or current SQL Server wait type.
Wait Resource	The SQL Server's textual representation of a lock resource.

I/O Statistics

The table below describes the statistics in the I/O category on the Overview tab of the Session Detail view:

Statistic	Description
Physical I/O	The number of physical and logical reads performed by this session.
Physical Reads	The number of physical reads from disk performed by this session.
Logical Reads	The number of reads from memory performed by this process.
Logical Writes	The number of writes to memory performed by this process.

Users Statistics

The table below describes the statistics in the Users category on the Overview tab of the Session Detail view:

Statistic	Description
CPU Usage	The cumulative CPU time for the process. The CPU usage is updated regardless of the value of the SET STATISTICS TIME ON option.
Open Trans	The current number of open transactions owned by the process.
Login Time	For client process, the last time a client logged into the SQL Server instance. For system processes, the time that SQL Server was started.

Statistic	Description
Last Batch	For client process, the last time a remote stored procedure call or an EXECUTE statement was executed. For system processes, the time that SQL Server was started.

Network Statistics

The table below describes the statistics in the Network category on the Overview tab of the Session Detail view:

Statistic	Description
Net Address	The unique identifier of the network card in the client machine that owns the process.
Net Library	When a process is initiated from a client, the controlling mechanism is the network connection. Each network connection has a library associated with it. This value is the name library associated with the network connection responsible for this process.

Metrics

High memory usage and a low cache hit ratio for a given process over a sustained period of time could indicate that the process is using poorly written code. Check the SQL tab to investigate further.

Also, watch for an unusually high percentage of CPU use over a long period of time. This could indicate a rogue process that must be terminated by the DBA using a KILL command for the session.

SQL Tab for SQL Server

- [Metrics](#)

The SQL tab of the Session Detail view presents information relating to any current SQL issued by the session as well as any SQL previously issued by the session.

Metrics

To determine access paths, you should export and run through a QUERY PLAN session, any SQL suspect of inefficient access.

Blocked By Tab for SQL Server

- [Metrics](#)
- [Troubleshooting](#)

Without a doubt, blocking lock situations can give the appearance of a “frozen” database almost more than anything else. A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although SQL Server supports row-level locking, blocking lock situations do crop up - sometimes frequently.

Metrics

The Blocked By tab contains information relating to processes that are currently blocking the process featured in the Session Detail view. The table below describes the information available on the Blocked By tab of the Session Detail view for SQL Server:

Column	Description
SPID	The SQL Server process ID. It is a unique value across all processes.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Login	The SQL Server login name of the process.
NT User	If using Windows Authentication, the name of the Windows NT user for this process. If using a trusted connection, the Windows NT user name.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Memory	The number of pages in the procedure cache (SQL Cache) that are currently allocated to this process. A negative number indicates that pages are being released (freed) from this process.
CPU	The cumulative CPU time for the process. The CPU usage is updated regardless of the value of the SET STATISTICS TIME ON option.
Physical I/O	The current cumulative number of physical disk reads and writes issued by the process.
Blocked	If the process is being blocked, the SPID of the blocking process.
Database	The database in which the process is running.
Command	The current command being executed by the process.
Last Batch	For a client process, the last time a remote stored procedure call or an EXECUTE statement was executed. For system processes, the time that SQL Server was started.
Login Time	For a client process, the last time a client logged into the SQL Server instance. For system processes, the time that SQL Server was started.
Wait Time	The time that the process has been waiting to be serviced, in milliseconds. A value of zero indicates that the process is not waiting.
Wait Type	The last or current SQL Server wait type.
Open Xacts	The current number of open transactions owned by the process.
NT Domain	If using Windows Authentication or a trusted connection, the name of the Windows domain of the user who owns the process.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

Blocking Tab for SQL Server

- [Metrics](#)
- [Troubleshooting](#)

Without a doubt, blocking lock situations can give the appearance of a “frozen” database almost more than anything else. A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems. Although SQL Server supports row-level locking, blocking lock situations do crop up - sometimes frequently.

Metrics

The Blocking tab contains information on blocks issued by the featured process that are blocking other processes. The table below describes the information available on the Blocking tab of the Session Detail view for SQL Server:

Column	Description
SPID	The SQL Server process ID. It is a unique value across all processes.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Login	The SQL Server login name of the process.
NT User	If using Windows Authentication, the name of the Windows NT user for this process. If using a trusted connection, the Windows NT user name.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Memory	The number of pages in the procedure cache (SQL Cache) that are currently allocated to this process. A negative number indicates that pages are being released (freed) from this process.
CPU	The cumulative CPU time for the process. The CPU usage is updated regardless of the value of the SET STATISTICS TIME ON option.
Physical I/O	The current cumulative number of physical disk reads and writes issued by the process.
Database	The database in which the process is running.
Command	The current command being executed by the process.
Last Batch	For client process, this value represents the last time a remote stored procedure call or an EXECUTE statement was executed. For system processes, it represents the time at which SQL Server was started.
Login Time	For client process, the last time a client logged into the SQL Server instance. For system processes, the time that SQL Server was started.
Wait Time	The time that the process has been waiting to be serviced, in milliseconds. A value of zero indicates that the process is not waiting.
Wait Type	The last or current SQL Server wait type.
Open Xacts	The current number of open transactions owned by the process.
NT Domain	If using Windows Authentication or a trusted connection, the name of the Windows Domain of the user who owns the process.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

All Locks Tab for SQL Server

- [Metrics](#)
- [Troubleshooting](#)

Without a doubt, blocking lock situations can give the appearance of a “frozen” database almost more than anything else. A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems. Although SQL Server supports row-level locking, blocking lock situations do crop up - sometimes frequently.

Metrics

The All Locks tab contains information on all locks associated with the featured SPID, including locks that are held by the process and locks that are blocking the process. The table below describes the information available on the All Locks tab of the Session Detail view for SQL Server:

Column	Description
Database	The database where the locks are occurring.
Table Name	The name of the table involved in a lock. NULL for non-table locks or table locks that take place in the tempdb database.
Ndx ID	The index ID involved in the lock.
Lock Type	The type of lock (database, table, row ID, etc.).
Lock Mode	The mode of the lock (shared, exclusive, etc.).
Lock Status	The status of the lock (waiting or granted).
Owner Type	Whether the lock came from a regular session or a transaction.
Program	The executable the process is using against the server.
BLK SPID	If nonzero, the process ID of the process blocking the requested lock. A value of zero indicates that the process is not blocked.
Wait Time	The time the process has waited for the lock, in milliseconds.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Command	The command the process is currently issuing.
NT Domain	The name of the Windows 2000/NT domain.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

By default, all processes wait indefinitely for locks in SQL Server. You can change this behavior by using the SET LOCK_TIMEOUT command, which limits the number of seconds that a process waits for a lock before timing out.

Sybase Session Detail View

The Session Detail view down provides a granular look at the details when a process is acting in a way that merits further investigation. It also presents data about a particular process in a way that makes it easier to understand and view apart from all other processes that Sybase is running.

The following tabbed pages are available on the Session Detail view for Sybase:

- [Overview](#)
- [SQL](#)
- [Blocked By](#)
- [Blocking](#)
- [All Locks](#)

Overview Tab for Sybase

The Overview tab of the Session Detail view displays information to analyze the details of a particular process. The tables below describe the statistics for each category on the Overview tab of the Session Detail view for Sybase. The available categories are:

- [Contention](#)
- [Execution](#)
- [General](#)
- [I/O](#)
- [Memory](#)
- [Users](#)

General Statistics

The table below describes the statistics in the General category on the Overview tab of the Session Detail view:

Statistic	Description
SPID	The Sybase process ID. Unique value across all processes.
Login Name	The Sybase login name of the process.
Family ID	The ID of the coordinating process and all of its worker processes.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Database	The database in which the process is running.
Program	The executable the process is using against the server.

Statistic	Description
Host	The machine name that originated the process.

Memory Statistics

The Memory Usage statistic in the Memory category on the Overview tab of the Session Detail view is the number of pages in the procedure cache (SQL Cache) that are currently allocated to this process. A negative number indicates that pages are being released (freed) from this process.

Contention Statistics

The table below describes the statistics in the Contention category on the Overview tab of the Session Detail view:

Statistic	Description
Blocked By	If the process is being blocked, the SPID of the blocking process. A value of zero indicates that the process is not blocked.
Time Blocked	The SQL Server's wait time.

I/O Statistics

The Physical I/O statistic in the I/O category on the Overview tab of the Session Detail view is the number of physical and logical reads performed by this session.

Users Statistics

The table below describes the statistics in the Users category on the Overview tab of the Session Detail view:

Statistic	Description
CPU Usage	The cumulative CPU time for the process. The CPU usage is updated regardless of the value of the SET STATISTICS TIME ON option.
Active Trans	The current number of open transactions owned by the process.

Execution Statistics

The table below describes the statistics in the Execution category on the Overview tab of the Session Detail view:

Statistic	Description
Exec Class	The execution class for the current process.
Priority	The priority of the current process.
Affinity	The affinity level for the current process.

SQL Tab for Sybase

- [Metrics](#)

The SQL tab of the Session Detail view presents information relating to any current SQL issued by the session as well as any SQL previously issued by the session.

Metrics

To determine access paths, you should export and run through a QUERY PLAN session, any SQL suspect of inefficient access.

Blocked By Tab for Sybase

- [Metrics](#)
- [Troubleshooting](#)

Without a doubt, blocking lock situations can give the appearance of a “frozen” database almost more than anything else. A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches even on large systems.

Metrics

The Blocked By tab contains information relating to processes that are currently blocking the process featured in the Session Detail view. The table below describes the information available on the Blocked By tab of the Session Detail view for Sybase:

Column	Description
SPID	The Sybase process ID. This is a unique value across all processes.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Login	The Sybase login name of the process.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Memory	The number of pages in the procedure cache (SQL Cache) that are currently allocated to this process. A negative number indicates that pages are being released (freed) from this process.
CPU	The cumulative CPU time for the process. The CPU usage is updated regardless of the value of the SET STATISTICS TIME ON option.
Physical I/O	The current cumulative number of physical disk reads and writes issued by the process.
Database	The database in which the process is running.
Command	The current command being executed by the process.
Time Blocked	The time that the process has been blocked, in seconds.
Transaction	The name of any transaction.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

Blocking Tab for Sybase

- [Metrics](#)
- [Troubleshooting](#)

Without a doubt, blocking lock situations can give the appearance of a “frozen” database almost more than anything else. A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems.

Metrics

The Blocking tab contains information on blocks issued by the featured process that are blocking other processes. The table below describes the information available on the Blocking tab of the Session Detail view for Sybase:

Column	Description
SPID	The Sybase process ID. This is a unique value across all processes.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Login	The Sybase login name of the process.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Memory	The number of pages in the procedure cache (SQL Cache) that are currently allocated to this process. A negative number indicates that pages are being released (freed) from this process.
CPU	The cumulative CPU time for the process. The CPU usage is updated regardless of the value of the SET STATISTICS TIME ON option.
Physical I/O	The current cumulative number of physical disk reads and writes issued by the process.
Database	The database in which the process is running.
Command	The current command being executed by the process.
Time Blocked	The time that the process has been blocked, in seconds.
Transaction	The name of any transaction.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied - the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects they were accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

All Locks Tab for Sybase

- [Metrics](#)
- [Troubleshooting](#)

Without a doubt, blocking lock situations can give the appearance of a “frozen” database almost more than anything else. A single blocking user has the potential to stop work for nearly all other processes on a small system, and can cause major headaches on large systems.

Metrics The All Locks tab contains information on all locks associated with the featured SPID, including locks that are held by the process and locks that are blocking the process. The table below describes the information available on the All Locks tab of the Session Detail view for Sybase:

Column	Description
Database	The database where the locks are occurring.
Lock Type	The type of lock (database, table, row ID, etc.)
Object Name	The name of the object being blocked.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Lock Page	The page number where the lock is currently applied.
Lock Class	The name of the cursor the lock is associated with (if any).
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Command	The command the process is currently issuing.
Transaction	The name of any transaction.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Troubleshooting

Once discovered, a blocking lock situation can normally be quickly remedied—the DBA issues a KILL against the offending process, which eliminates the user's stranglehold on the objects the user was accessing. Other user processes then nearly almost always complete in an instant. Discovering the blocked lock situation is made easier by using tools like Performance Center, but preventing the blocking lock situation in the first place is tricky.

The culprit of blocking lock scenarios is usually the application design, or the SQL being used within the application itself. Properly coding an application to reference database objects in an efficient order, and then using the right SQL to get the job done, is an art. The key to avoiding lock contention is to process user transactions in the quickest and most efficient manner possible - something not always easy to do.

DB2 Session Detail View

The following tabbed pages are available on the Session Detail view for DB2:

- [Application Tab for DB2](#)
- [Unit of Work Tab for DB2](#)
- [Locking Tab for DB2](#)
- [Memory Tab for DB2](#)
- [I/O Tab for DB2](#)
- [SQL Statistics Tab for DB2](#)

The following information about the selected application is available on each page:

Field	Description
Agent ID	This is a system-wide unique identifier for the application.
Auth ID	This is the authorization ID of the user who invoked the application being monitored.

Field	Description
OS User ID	This is the authorization ID used to access the operating system.
Application	This is the name of the application executable.

Application Tab for DB2

The Application tab contains application and client detail information from the application snapshot record for the selected application.

Application Details

Statistic	Description
Application Agent ID	This is a system-wide unique identifier for the application.
Application Agent PID	The process ID (UNIX) or thread ID (Windows) of a DB2 agent.
Application Agent System CPU (sec)	This is the total system CPU time (in seconds) used by the database manager agent process.
Application Agent User CPU (sec)	This is the total CPU time (in seconds) used by database manager agent process.
Application Agents	This is the number of agents currently executing statements or subsections.
Application Agents Stolen	This is the number of times agents are stolen from the application. Agents are stolen when an idle agent is reassigned from one application to another.
Application Assoc Agents HWM	This is the maximum number of subagents associated with the application.
Application Assoc Sub-Agents	This is the number of subagents associated with the application.
Application Authorization ID	This is the authorization ID of the user who invoked the application being monitored.
Application Codepage	This is the code page identifier.
Application Connect Complete Time	This is the date and time a connection request was granted.
Application Connect Start Time	This is the date and time an application started a connection request.
Application Coord Agent PID	This is the process ID (UNIX) or thread ID (Windows) of the coordinator agent for the selected application.
Application Coord Node	In a multi-node system, this is the node number of the node where the selected application connected or attached to the instance.
Application Coord Token	This is the DRDA AS correlation token.
Application ID	This is generated when the application connects to the database at the database manager or when DDCS receives a request to connect to a DRDA database.
Application Idle Time	This is the number of seconds since the selected application last issued any requests to the server. This includes applications that have not terminated a transaction, for example not issued a commit or rollback.
Application Name	Name of the application executable.
OS User ID	The authorization ID used to access the operating system.
Application Priority	This is the priority of the agents working for this application.
Application Priority Type	This is the operating system priority type for the agent working on behalf of the application.
Application Sequence Number	This identifier is incremented when a unit of work ends (when a COMMIT or ROLLBACK terminates a unit of work). Together, the application ID and application sequence number uniquely identify a transaction.

Statistic	Description
Application Session Auth ID	This is the current authorization ID for the session being used by this application.
Application Status	This is the current status of the application.
Application Status Change Time	This is the date and time the application entered its current status.
Application Territory Code	This is the territory code (formerly country code) of the database for which the monitor data is being collected.

Application Authorities

Column	Description
Authority	This is the highest authority level granted to the application.
Explicit	Authorizations granted explicitly to a user.
Indirect	Indirect authorizations inherited from group or public.

Client Details

Statistic	Description
Client Database Alias	This is the alias of the database provided by the application to connect to the database.
Client Inbound Comm Address	This is the communication address of the client.
Client Node Name	This is the node name (nname) in the database manager configuration file at the client database partition.
Client PID	This is the process ID of the client application that made the connection to the database.
Client Platform	This is the operating system on which the client application is running.
Client Product and Version	This is the product and version that is running on the client.
Client Protocol	This is the communication protocol that the client application is using to communicate with the server.
TP Client Accounting String	This is the data passed to the target database for logging and diagnostic purposes (if the sqleseti API was issued in this connection).
TP Client Application Name	This name identifies the server transaction program performing the transaction (if the sqleseti API was issued in this connection).
TP Client User ID	This is the client user ID generated by a transaction manager and provided to the server (if the sqleseti API is used).
TP Client Workstation Name	This name identifies the client's system or workstation, for example CICS EITERMID, if the sqleseti API was issued in this connection.

Unit of Work Tab for DB2

The Unit of Work tab contains the SQL statement for the selected application, as well as statistics related to that statement.

SQL Statement

The SQL Statement field displays the SQL statement for the application. You can click **Explain SQL** to view an explain plan for the statement.

Statement Detail

Statistic	Description
SQL Statement	This is the text of the dynamic SQL statement.
Node	This is the node where the statement was executed.
Type	This is the type of statement processed.
Creator	This is the authorization ID of the user that pre-compiled the application.
Operation	This is the statement operation currently being processed or that was most recently processed (if none are currently running).
Agents Created	This is the maximum number of agents that were used when executing the statement.
Agents Working	This is the number of concurrent agents currently executing a statement or subsection.
Cursor Name	This is the name of the cursor corresponding to this SQL statement.
Blocking Cursor	This indicates if the statement being executed is using a blocking cursor.
Package Name	This is the name of the package that contains the SQL statement that is currently executing.
Package Version	This identifies the version identifier of the package that contains the currently executing SQL statement.
Section Number	This is the internal section number in the package for the SQL statement currently processing or most recently processed.
Parallelism Degree	This is the degree of parallelism requested when the query was bound.
Query Cost Estimate	This is the estimated cost for a query (in timerons) as determined by the SQL compiler.
Query Card Estimate	This is an estimate of the number of rows that will be returned by a query.

Statement Statistics

Statistic	Description
Statement Start Time	This is the date and time when the statement operation (stmt_operation) monitor element started executing.
Statement Stop Time	This is the date and time when the statement operation (stmt_operation) monitor element stopped executing.
Statement Sort Time (sec)	This is the total elapsed time (in seconds) for all sorts that have been executed.
Statement User CPU (sec)	This is the total user CPU time (in seconds) used by the currently executing statement.
Statement System CPU (sec)	This is the total system CPU time (in seconds) used by the currently executing statement.
Statement Elapsed Time (sec)	This is the elapsed execution time (in seconds) of the most recently completed statement.
UOW Start Time	This is the date and time that the unit of work first required database resources.
UOW Stop Time	This is the date and time that the most recent unit of work completed. This occurs when database changes are committed or rolled back.
UOW Prev Stop Time	This is the time the previous unit of work completed.
UOW Elapsed Time (sec)	This is the elapsed execution time (in seconds) of the most recently completed unit of work.
UOW Lock Wait Time (sec)	This is the total amount of elapsed time (in seconds) this unit of work has spent waiting for locks.
UOW Log Space Used (KB)	This is the amount of log space (in kilobytes) used in the current unit of work of the monitored application.
UOW Completion Status	This is the status of the unit of work and how it stopped.

Statement Activity

Statistic	Description
Sorts	This is the total number of times a set of data was sorted in order to process the statement operation (stmt_operation).
Sort Overflows	This is the total number of sorts that ran out of sort heap and may have used disk space for temporary storage.
Fetches	This is the number of successful fetches that were performed on a specific cursor.
Rows Read	This is the number of rows read from the table.
Rows Written	This is the number of rows changed (inserted, deleted, or updated) in the table.
Internal Rows Deleted	This is the number of rows deleted from the database as a result of internal activity.
Internal Rows Updated	This is the number of rows updated from the database as a result of internal activity.
Internal Rows Inserted	This is the number of rows inserted into the database as a result of internal activity that was caused by triggers.
Temporary Data Logical Reads	This indicates the number of data pages that have been requested from the buffer pool (logical) for temporary table spaces. NOTE: The functionality to record buffer pool information at the statement level is supported for API and CLP snapshot requests.
Temporary Data Physical Reads	Indicates the number of data pages that have been read in from the table space containers (physical) for temporary table spaces. NOTE: The functionality to record buffer pool information at the statement level is supported for API and CLP snapshot requests.
Temporary Index Logical Reads	Indicates the number of index pages that have been requested from the buffer pool (logical) for temporary table spaces. NOTE: The functionality to record buffer pool information at the statement level is supported for API and CLP snapshot requests.
Temporary Index Physical Reads	Indicates the number of index pages that have been read in from the table space containers (physical) for temporary table spaces. NOTE: The functionality to record buffer pool information at the statement level is supported for API and CLP snapshot requests.

Locking Tab for DB2

The Locking tab is a single application view of the [Lock View](#). Information that is found in the Locks Held and Locks Waiting tabs of the Lock View will be available here for the selected application.

Locks Held

Column	Description
Lock Mode	This is the type of lock being held.
Object Type	This is the type of object against which the application holds a lock (for object-lock-level information), or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information).
Table Schema	This is the schema of the table that the lock is on.
Table Name	This is the name of the table that the lock is on. This element is only set if Object Type indicates Table.
Tablespace	This is the name of the table space against which the lock is held.
Lock Status	This is the lock's status (waiting or granted).

Column	Description
Escalation	This indicates whether a lock request was made as part of a lock escalation.

Locks Waiting

Column	Description
Agent ID	This is the application handle of the agent holding a lock for which this application is waiting.
Application ID	This is the application ID of the application that is holding a lock on the object that this application is waiting to obtain.
Lock Mode	This is the type of lock being held.
Mode Requested	This is the lock mode being requested by the application.
Object Type	This is the type of object against which the application holds a lock.
Table Schema	This is the schema of the table that the lock is on.
Tablespace	This is the name of the table that the lock is on. This element is only set if Object Type indicates Table.
Wait Start Time	This is the date and time that this application started waiting to obtain a lock on the object that is currently locked by another application.
Escalation	This indicates whether a lock request was made as part of a lock escalation.

Memory Tab for DB2

The Memory tab displays all the memory pools for the application and key cache statistics specific to the application.

Memory Pools

Column	Description
Node	This is the number assigned to the node in the db2nodes.cfg file.
Memory Pool	This is the type of memory pool.
Utilization	This is the percentage of memory pool used.
High Watermark (MB)	This is the largest size of a memory pool (in megabytes) since its creation.
Current Size (MB)	This is the current size of a memory pool (in megabytes).
Max Size (MB)	This is the internally configured size of a memory pool (in megabytes) in DB2.

Memory Statistics

Statistic	Description
Application Section Inserts	This is the number of inserts of SQL sections by an application from its SQL work area.
Application Section Lookups	This is the number of lookups of SQL sections by an application from its SQL work area.
Catalog Cache Inserts	This is the number of times that the system tried to insert table descriptor or authorization information into the catalog cache.
Catalog Cache Lookups	This is the number of times that the catalog cache was referenced to obtain table descriptor information or authorization information.
Catalog Cache Overflows	This is the number of times that the catalog cache overflowed the bounds of its allocated memory.

Statistic	Description
Package Cache Inserts	This is the total number of times that a requested section was not available for use and had to be loaded into the package cache. This includes any implicit prepares performed by the system.
Package Cache Lookups	This is the number of times an application looked for a section or package in the package cache.
Private Workspace Inserts	This is the number of inserts of SQL sections by an application into the private workspace.
Private Workspace Lookups	This is the number of lookups of SQL sections by an application in its agents' private workspace.
Private Workspace Overflows	This is the number of times that the private workspaces overflowed the bounds of its allocated memory.
Private Workspace HWM (KB)	This is the largest size (in kilobytes) reached by the private workspace.
Shared Workspace Inserts	This is the number of inserts of SQL sections by applications into shared workspaces.
Shared Workspace Lookups	This is the number of lookups of SQL sections by applications in shared workspaces.
Shared Workspace Overflows	This is the number of times that shared workspaces overflowed the bounds of their allocated memory.
Shared Workspace HWM (KB)	This is the largest size (in kilobytes) reached by shared workspaces.

Cache Hit Ratios

Ratio	Description
Catalog Cache	The catalog cache hit ratio indicates how well the catalog cache is helping to avoid actual accesses to the catalog on disk. A high ratio indicates successful avoidance of actual disk I/O accesses.
Package Cache	The package cache hit ratio indicates how well the package cache is helping to avoid reloading packages and sections for static SQL from the system catalogs as well as helping to avoid recompiling dynamic SQL statements. A high ratio indicates successful avoidance of these activities.
Shared Workspace	The shared workspace hit ratio indicates how well the shared SQL workspace is helping to avoid having to initialize sections for SQL statements that are about to be executed. A high ratio indicate successful avoidance of this action.
Private Workspace	The private workspace hit ratio indicates how well the private SQL workspace is helping to avoid having to initialize sections for SQL statements that are about to be executed. A high ratio indicate successful avoidance of this action.
Application Workspace	The application workspace hit ratio indicates how well the application SQL workspace is helping to avoid having to initialize sections for SQL statements that are about to be executed. A high ratio indicate successful avoidance of this action.

I/O Tab for DB2

The I/O tab displays detailed I/O information from the application snapshot record for the selected application.

I/O Statistics

Statistic	Description
Buffered Data Logical Reads	This indicates the number of data pages that have been requested from the buffer pool (logical) for regular and large table spaces.

Statistic	Description
Buffered Data Physical Reads	This indicates the number of data pages that have been read from the table space containers (physical) for regular and large table spaces.
Buffered Data Writes	This indicates the number of times a buffer pool data page was physically written to disk.
Buffered Index Logical Reads	This indicates the number of index pages that have been requested from the buffer pool (logical) for regular and large table spaces.
Buffered Index Physical Reads	This indicates the number of index pages that have been read from the table space containers (physical) for regular and large table spaces.
Buffered Index Writes	This indicates the number of times a buffer pool index page was physically written to disk.
Buffered Read Time (sec)	This indicates the total amount of time (in seconds) spent reading data and index pages from the table space containers (physical) for all types of table spaces.
Buffered Write Time (sec)	This indicates the total amount of time (in seconds) spent physically writing data or index pages from the buffer pool to disk.
Direct Reads	This is the number of read operations that do not use the buffer pool.
Direct Read Requests	This is the number of requests to perform a direct read of one or more sectors of data.
Direct Writes	This is the number of write operations that do not use the buffer pool.
Direct Write Requests	This is the number of requests to perform a direct write of one or more sectors of data.
Direct Read Time (sec)	This is the elapsed time (in seconds) required to perform the direct reads.
Direct Write Time (sec)	This is the elapsed time (in seconds) required to perform the direct writes.
Extended Storage – Data Pages From	This is the number of buffer pool data pages copied from extended storage.
Extended Storage – Data Pages To	This is the number of buffer pool data pages copied to extended storage.
Extended Storage – Index Pages From	This is the number of buffer pool index pages copied from extended storage.
Extended Storage – Index Pages To	This is the number of buffer pool index pages copied to extended storage.
Prefetch Pages Unread	This indicates the number of pages that the prefetcher read in that were never used.
Prefetch Wait Time (sec)	This is the time an application spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool.
Temporary Data Logical Reads	This indicates the number of data pages which have been requested from the buffer pool (logical) for temporary table spaces.
Temporary Data Physical Reads	This indicates the number of data pages read in from the table space containers (physical) for temporary table spaces.
Temporary Index Logical Reads	This indicates the number of index pages which have been requested from the buffer pool (logical) for temporary table spaces.
Temporary Index Physical Reads	This indicates the number of index pages read in from the table space containers (physical) for temporary table spaces.

I/O Distribution Ratios

Ratio	Description
Direct Read Ratio	Direct Reads are read operations that do not use the buffer pool. Direct Read Ratio is the percentage of all reads that were direct reads.

Ratio	Description
Logical Read Ratio	Logical Reads is the sum of all Buffer Pool Data Logical Reads and Buffer Pool Index Logical Reads. Logical Read Ratio is the percentage of all reads that were logical reads.
Physical Read Ratio	Physical Reads is the sum of all Buffer Pool Data Physical Reads and Buffer Pool Index Physical Reads. Physical Read Ratio is the percentage of all reads that were physical reads.
Direct Write Ratio	Direct Writes are write operations that do not use the buffer pool. Direct Write Ratio is the percentage of all writes that were direct writes.
Buffered Write Ratio	Buffered Writes is the sum of all Buffer Pool Data Writes and Buffer Pool Index Writes. Buffered Write Ratio is the percentage of all writes that were buffered writes.

SQL Statistics Tab for DB2

The SQL Statistics tab contains statistics about the SQL statement for the selected application.

SQL Statistics

Statistic	Description
Binds and Pre-Compiles	This is the number of binds and pre-compiles attempted.
Cursor Block Requests Accepted	This is the number of times a request for an I/O block was accepted.
Cursor Block Requests Rejected	This is the number of times a request for an I/O block at the server was rejected and the request was converted to non-blocked I/O.
Cursors Open Local	This is the number of local cursors currently open for this application, including those cursors counted by Cursors Open Local with Blocking.
Cursors Open Local with Blocking	This is the number of local blocking cursors currently open for this application.
Cursors Open Remote	This is the number of remote cursors currently open for this application, including those cursors counted by Cursors Open Remote with Blocking.
Cursors Open Remote with Blocking	This is the number of remote blocking cursors currently open for this application.
Internal Automatic Rebinds	This is the number of automatic rebinds (or recompiles) that have been attempted.
Internal Commits	This is the total number of commits initiated internally by the database manager.
Internal Rollbacks	This is the total number of rollbacks initiated internally by the database manager.
Internal Deadlock Rollbacks	This is the total number of forced rollbacks initiated by the database manager due to a deadlock. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.
Internal Rows Deleted	This is the number of rows deleted from the database as a result of internal activity.
Internal Rows Inserted	This is the number of rows inserted into the database as a result of internal activity caused by triggers.
Internal Rows Updated	This is the number of rows updated from the database as a result of internal activity.
Rows Deleted	This is the number of row deletions attempted.
Rows Inserted	This is the number of row insertions attempted.
Rows Read	This is the number of rows read from the table.
Rows Selected	This is the number of rows that have been selected and returned to the application.
Rows Updated	This is the number of row updates attempted.
Rows Written	This is the number of rows changed (inserted, deleted, or updated) in the table.

Statistic	Description
SQL DDL Statements	This indicates the number of SQL Data Definition Language (DDL) statements that were executed.
SQL Commit Statements	This indicates the total number of SQL COMMIT statements that have been attempted.
SQL Dynamic Statements	This indicates the number of dynamic SQL statements that were attempted.
SQL Failed Statements	This indicates the number of SQL statements that were attempted and failed.
SQL Requests Since Last Commit	This indicates the number of SQL requests submitted since the last commit.
SQL Rollback Statements	This indicates the total number of SQL ROLLBACK statements that have been attempted.
SQL Select Statements	This indicates the number of SQL SELECT statements that were executed.
SQL Static Statements	This indicates the number of static SQL statements that were attempted.
SQL UID Statements	This indicates the number of SQL UPDATE, INSERT, and DELETE statements that were executed.

SQL Distribution Ratios

Column	Description
DDL Statements	This is the percentage of all executed statements that were SQL DDL Statements.
UID Statements	This is the percentage of all executed statements that were SQL UID Statements.
Failed Statements	This is the percentage of all executed statements that were SQL Failed Statements.
Select Statements	This is the percentage of all executed statements that were SQL Select Statements.

Top Sessions View

The following tabbed pages are available on the Top Sessions view:

- [Memory Tab](#)
- [I/O Tab](#)
- [CPU Tab](#)

Memory Tab

The information on the Memory tab of the Top Sessions view depends on the target DBMS:

- [Memory Tab for Oracle](#)
- [Memory Tab for SQL Server](#)
- [Memory Tab for Sybase](#)
- [Memory Tab for DB2](#)

Memory Tab for Oracle

- [Metrics](#)

It is frequently the case that one or two users cause the majority of run-time problems. The problem could originate with a runaway process, an untuned batch procedure, or other user-initiated operation. User connections can get out of hand with memory consumption, and extreme cases have caused difficulties at both the database and operating system levels (ORA-4030 errors).

The table below describes the information available on the Leading Sessions tab of the Memory Detail view and the Memory tab of the Top Sessions view for Oracle:

Column	Description
User Name	The logon name the session is using.
SID	The unique Oracle identifier for the session.
Serial #	The serial number assigned to the session.
Status	The status of the session, ACTIVE or INACTIVE.
Machine	The name of the client machine name that the session is using.
PGA Memory (KB)	The Program Global Area (PGA) is a private memory area devoted to housing the global variables and data structures for a single Oracle process, in KB.
UGA Memory (KB)	The User Global Area (UGA) contains session specific information regarding open cursors, state information for packages, database link information, and more. When using Oracle's Multi-threaded Server (MTS), the UGA can be moved up into the SGA, in KB.
Memory Sorts (KB)	The total number of memory sorts a session has performed.
Total Memory (KB)	The memory (PGA + UGA) that the session is consuming, in KB.

NOTE: This information is available on both the [Leading Sessions tab](#) of the Memory Detail view and the Memory tab of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If your database server does not have an overabundance of memory, you should check periodically to see who your heavy memory users are and the total percentage of memory each takes up. If you see that one or two users use more than 5-15 percent of the total memory, you should investigate the sessions further to see what activities they are performing.

Memory Tab for SQL Server

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. User connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system levels.

Performance Center displays information to find processes that are using the most memory on the server on the Leading Sessions tab of the Memory Detail view and the Memory tab of the Top Sessions view.

The table below describes the information available on these tabs:

Column	Description
PID	The process ID of the connected session.
User Name	The user name assigned to the process.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The machine name that originated the process.
Program	The program the process has invoked against SQL Server.
Memory Usage	The total number of memory pages allocated to the process.
Pct Mem Used	The percentage of overall memory among all processes that can be attributed to the process.
Database	The database in which the process is currently running.
Command	The command the process is currently issuing.
Open Trans	The number of open transactions for the process.
Blocked	The PID of any process blocking the current process.
Wait Time	The current wait time for the process, in milliseconds.

NOTE: This information is available on both the [Leading Sessions tab](#) of the Memory Detail view and the Memory tab of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If your database server does not have an overabundance of memory, you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 5-15 percent of the total memory usage, you should investigate the sessions further to see what activities they are performing.

Memory Tab for Sybase

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Often, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system levels.

This tab displays information to find processes that are using the most memory on the server. The table below describes the information available on the Leading Sessions tab of the Memory Detail view and the Memory tab of the Top Sessions view for Sybase:

Column	Description
PID	The process ID.
User Name	The user name assigned to the process.
FID	The process ID of the worker process' parent.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Memory Usage	The memory currently used by the process.
Pct Mem Used	The percentage of memory currently used by the process.
Database	The database in which the process is running.
Command	The command the process is currently issuing.
Transaction	The name of any transaction.
Blocked	The PID of any process blocking the current process.
Time Blocked	The time that the process has been blocked, in seconds.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If your database server does not have an overabundance of memory, you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 5-15 percent of the total memory usage, you should investigate the sessions further to see what activities they are performing.

Memory Tab for DB2

- [Metrics](#)

It is not uncommon for one or two users to cause the majority of runtime problems that plague a database. The problem could be a runaway process, an untuned batch procedure, or other user-initiated operation. Often, user connections can get out of hand with memory consumption, and extreme cases have caused headaches at both the database and operating system levels.

This tab displays information to find processes that are using the most memory on the server. The table below describes the information DB2 on the Leading Sessions tab of the Memory Detail view and the Memory tab of the Top Sessions view for DB2:

Column	Description
Agent ID	The application handle of the agent holding a lock for which this application is waiting.
Auth ID	The authorization ID of the user who invoked the application that is being monitored.
OS User ID	The operating system ID of the process.
Client PID	The process ID of the client application that made the connection to the database.
Application	The name of the application executable.
Status	The current status of the application.
UOW Elapsed Time (sec)	The elapsed execution time of the most recently completed unit of work.
Memory Overflows	Total number of memory overflows.
Memory Used (KB)	Total memory pool usage for the application.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

If your database server does not have an overabundance of memory, you should periodically check to see who your heavy memory users are along with the total percentage of memory each takes up. If you see one or two users who have more than 5-15 percent of the total memory usage, you should investigate the sessions further to see what activities they are performing.

I/O Tab

The information on the I/O tab of the Top Sessions view depends on the target DBMS:

- [I/O Tab for DB2](#)
- [I/O Tab for SQL Server](#)
- [I/O Tab for Sybase](#)
- [I/O Tab for DB2](#)

I/O Tab for Oracle

- [Metrics](#)

When a system undergoes heavy I/O activity, all the user connections can contribute somewhat equally to the overall load. Frequently, however, one or two user connections are responsible for 75 percent or more of the I/O activity. It may be that a large batch load or other typical process is running, and that is perfectly okay for your system. Or, it may be a runaway process or rogue connection that you should track down and eliminate.

It is a good idea to periodically check the leading sessions in your system with respect to I/O and make sure all is well. You can use Performance Center to perform this function with the information available on this tab. The table below describes the information available on the Leading Sessions tab of the I/O Detail view and the I/O tab of the Top Sessions view for Oracle:

Column	Description
User Name	The logon name the session is using.
SID	The unique Oracle identifier for the session.
Serial #	The serial number assigned to the session.
Status	The status of the session, ACTIVE or INACTIVE.
Machine	The name of the client machine name that the session is using.
Reads	The number of physical reads.
Writes	The number of physical writes.
Total I/O	The total of all physical I/O operations for the session.

NOTE: This information is available on both the [Leading Sessions tab](#) of the I/O Detail view and the I/O tab of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Pinpointing sessions with abnormally high I/O activity relative to other sessions in the system can aid in ferreting out accounts that are dragging down overall system performance. You should examine the activity of each session to determine the system workload and to see if you can reduce the workload or tune the system for better performance.

I/O Tab for SQL Server

- [Metrics](#)

When a system undergoes heavy I/O activity, sometimes you find that all the user connections are contributing somewhat equally to the overall load. Frequently, however, one or two user connections are responsible for 75 percent or more of the I/O activity. It can be that a large batch load or other typical process is running that is perfectly okay for your system. Alternatively, it can be a runaway process or other rogue connection that you should track down and possibly eliminate.

Performance Center displays information to find processes that are using the most memory on the server on the Leading Sessions tab of the I/O Detail view and the I/O tab of the Top Sessions view.

The table below describes the information available on these tabs:

Column	Description
PID	The process ID.
User Name	The user name assigned to the process.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Physical I/O	The current cumulative number of reads and writes issued by the process.
Pct I/O Used	The percentage of overall I/O that can be attributed to the process.
Database	The database in which the process is running.
Command	The command the process is currently issuing.
Open Trans	The number of open transactions for the process.
Blocked	The PID of any process blocking the current process.
Wait Time	The current wait time for the process, in milliseconds.

NOTE: This information is available on both the [Leading Sessions tab](#) of the I/O Detail view and the I/O tab of the Top Sessions view.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Finding one more two users that are consuming more than 75 percent of the total I/O load can indicate a runaway or improper process. By drilling down into the I/O activity of all users, you can quickly see if this is the case.

I/O Tab for Sybase

- [Metrics](#)

When a system undergoes heavy I/O activity, sometimes you find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are responsible for 75 percent or more of the I/O activity. It can be that a large batch load or other typical process is running that is perfectly okay for your system. Or it can be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

It is a good idea to periodically check who the leading sessions are in your system with respect to I/O and make sure all is well. You can use Performance Center to easily perform this function with the leading sessions tab of the I/O Detail view. The table below describes the information available on the I/O tab of the Top Sessions view for Sybase:

Column	Description
PID	The process ID.
User Name	The user name assigned to the process.
FID	The process ID of the worker process' parent.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
Physical I/O	The current cumulative number of reads and writes issued by the process.
Pct I/O Used	The percentage of overall I/O that can be attributed to the process.
Database	The database in which the process is running.
Command	The command the process is currently issuing.
Transaction	The name of any transaction.
Blocked	The PID of any process blocking the current process.
Time Blocked	The time that the process has been blocked, in seconds.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Pinpointing sessions with abnormally high I/O activity relative to other sessions in the system help you ferret out accounts dragging down overall system performance. You should examine the activity of each session to determine the workload being placed on the system and if you can reduce or tune that workload for better performance.

I/O Tab for DB2

- [Metrics](#)

When a system undergoes heavy I/O activity, sometimes you find that all the user connections are contributing somewhat equally to the overall load. More often than not, however, one or two user connections are responsible for 75 percent or more of the I/O activity. It can be that a large batch load or other typical process is running that is perfectly okay for your system. Or it can be a runaway process or other rogue connection that needs to be tracked down and possibly eliminated.

It is a good idea to periodically check who the leading sessions are in your system with respect to I/O and make sure all is well. You can use Performance Center to easily perform this function with the leading sessions tab of the I/O Detail view. The table below describes the information available on the I/O tab of the Top Sessions view for DB2:

Column	Description
Agent ID	The application handle of the agent holding a lock for which this application is waiting.
Auth ID	The authorization ID of the user who invoked the application that is being monitored.
OS User ID	The operating system ID of the process.
Client PID	The process ID of the client application that made the connection to the database.
Application	The name of the application executable.
Status	The current status of the application.
UOW Elapsed Time (sec)	The elapsed execution time of the most recently completed unit of work.
Buffered I/O Time (ms)	The total time spent by application in performing buffered reads and writes.
Direct I/O Time (ms)	The total time spent by application in performing non-buffered reads and writes.
Total I/O Time (ms)	The total time spent by application in performing buffered and non-buffered reads and writes.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

Metrics

Pinpointing sessions with abnormally high I/O activity relative to other sessions in the system help you ferret out accounts dragging down overall system performance. You should examine the activity of each session to determine the workload being placed on the system and if you can reduce or tune that workload for better performance.

CPU Tab

The information on the CPU tab of the Top Sessions view depends on the target DBMS:

- [CPU Tab for Oracle](#)
- [CPU Tab for SQL Server](#)
- [CPU Tab for Sybase](#)
- [CPU Tab for DB2](#)

CPU Tab for Oracle

The table below describes the information available on the CPU tab of the Top Sessions view for Oracle:

Column	Description
User Name	The logon name the session is using.
SID	The unique Oracle identifier for the session.
Serial #	The serial number assigned to the session.
Status	The status of the session, ACTIVE or INACTIVE.
Machine	The name of the client machine name that the session is using.
Program	The executable the process is using against the server.
CPU	The CPU used by the process when the call started.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

CPU Tab for SQL Server

The table below describes the information available on the CPU tab of the Top Sessions view for SQL Server:

Column	Description
PID	The process ID.
User Name	The user name assigned to the process.
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
CPU	The cumulative CPU time for the process.
Pct CPU Used	The percentage of the CPU dedicated to this process.
Database	The database in which the process is running.
Command	The command the process is currently issuing.
Open Trans	The number of open transactions for the process.
Blocked	The PID of any process blocking the current process.
Wait Time	The current wait time for the process, in milliseconds.

TIP: To configure the grid to show/hide row numbers, use the [Options Editor](#).

CPU Tab for Sybase

The table below describes the information available on the CPU tab of the Top Sessions view for Sybase:

Column	Description
PID	The process ID.
User Name	The user name assigned to the process.
FID	The process ID of the worker process' parent.

Column	Description
Status	Indicates if the process is actively performing work, is idle, blocked by another process, etc.
Host	The machine name that originated the process.
Program	The executable the process is using against the server.
CPU	The cumulative CPU time for the process in ticks.
Pct CPU Used	The percentage of the CPU dedicated to this process.
Database	The database in which the process is running.
Command	The command the process is currently issuing.
Transaction	The name of any transaction.
Blocked	The PID of any process blocking the current process.
Time Blocked	The time that the process has been blocked, in seconds.

To configure the grid to show/hide row numbers, use the [Options Editor](#).

CPU Tab for DB2

The table below describes the information available on the CPU tab of the Top Sessions view for DB2:

Column	Description
Agent ID	The application handle of the agent holding a lock for which this application is waiting.
Auth ID	The authorization ID of the user who invoked the application that is being monitored.
OS User ID	The operating system ID of the process.
Client PID	The process ID of the client application that made the connection to the database.
Application	The name of the application executable.
Status	The current status of the application.
UOW Elapsed Time (sec)	The elapsed execution time of the most recently completed unit of work.
Agent ID	The application handle of the agent holding a lock for which this application is waiting.
User CPU Time (sec)	The total user CPU time used by the application agents.
System CPU Time (sec)	The total system CPU time used by the application agents.
Total CPU Time (sec)	The total user + system CPU time used by the application agents.

To configure the grid to show/hide row numbers, use the [Options Editor](#).

Top SQL View

- [Metrics](#)

One or two bad queries can cause a lot of trouble for the remaining sessions in a database. It is important to find them before they get into a production system, but sometimes a few sneak through.

By applying custom filters and performance-related thresholds, the Top SQL view locates inefficient SQL. By applying filters to certain I/O and statistical counters, you hope to isolate queries that far exceed their nearest competitors in the same area (like disk reads). When you find them, you should reduce the number of sorts a query performs. Or, for a query that returns only a few records, you should try to minimize the number of rows a query processes.

The Top SQL view displays requested SQL for SQL Server, Oracle, DB2, and Sybase datasources.

Metrics

When you begin to look for inefficient SQL in a database, there are two primary questions you need to answer:

- 1 What has been the worst SQL that has historically been run in my database?
- 2 What is the worst SQL that's running right now in my database?

When troubleshooting a slow system, you should be on the lookout for any query that shows an execution count that is significantly larger than any other query on the system. It may be that the query is in an inefficient Transact SQL loop, or other problematic programming construct. Only by bringing the query to the attention of the application developers will you know if the query is being mishandled from a programming standpoint.

Glossary

A

Action

A predefined response to a threshold violation.

Alarms View

An HTML window where you can see alarms, for all monitored datasources. In the Alarms view, you can set alarms for any threshold violation. When a violation occurs, Performance Center initiates an alarm and logs it for viewing.

Alert Logs View

An HTML window where you can see a list of your Oracle Alert Logs from the Performance Center application. It includes filters, which let you customize your display for specific Alert Logs. The Performance Center Alert Log view lets you view a list of your Oracle Alert.

Archive View

An HTML window where you can see lists of all archived log files written out by the database.

Analyzer

The Performance Center Server internal process that performs threshold validation.

Apache

A public-domain Web server developed by a loosely-knit group of programmers known as the Apache Group. Apache httpd Server is based on a popular, public-domain HTTP daemon developed in 1995 by Rob McCool at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign. Because it was developed from existing NCSA code and various patches, it was called “a patchy” server - hence the name “Apache”.

Performance Center uses Apache 2.2.6 to communicate between the Performance Center Server and Performance Center Clients.

B

Blackout

A schedule that is a time interval where Performance Center will not send notifications about any normal threshold violations or performance problems.

C

Capture Agent

The Performance Center Server internal process that retrieves values for a certain list of statistics and then writes them to the repository.

Category

Type of statistic.

Performance Center breaks up all of the statistics into eight categories, Memory, I/O, Space, Objects, Databases, Contention, Users, and Network. Each of these categories has a performance category view. All of the statistics for each category are available on the individual performance category views.

Clone

A copy of a datasource including all of the properties (thresholds, notifications etc.) from the original datasource.

D**Datasource Manager**

The Performance Center Server internal datasource connection pool manager.

DBWR

Database Writer Process.

The DBWR process handles the flow of information from Oracle physical datafiles to and from the various memory structures in the system global area (SGA).

E**Enterprise**

The network (or interconnected networks) of computer systems. It can span diverse geographical locations and usually encompasses a range of platforms, operating systems, protocols, and network architectures.

Enterprise View

An HTML window designed to give you a comprehensive picture of the overall health of all monitored datasources in your enterprise. It displays all currently registered datasources. The Enterprise view lets you know what databases are up and down and gives you other critical indicators regarding the status of your monitored datasources.

F

There are no entries for this letter.

G

There are no entries for this letter.

H

Health Index

The set of statistics used to define the various performance indexes used to globally measure database performance. The Health Index provides important statistics for monitoring the performance of datasources in an enterprise.

Health Index Template

Logical containers that define the various performance indexes used to globally measure database performance.

Home View

An HTML window where you can see the statistic category vital signs. The Home view is the default performance category view.

I

I/O

Input/Output.

Physical I/O is read to/from a disk. Logical I/O is read to/from memory.

Init.ora

The Oracle Initialization Parameter File. It is a text file containing a list of parameters and values for each parameter. You can modify many initialization parameters to improve database performance.

J

There are no entries for this letter.

K

There are no entries for this letter

L

LGWR

Log Writer Process.

The LGWR manages the information contained in Oracle online redo log files and redo log buffer area.

Login

A login is a form of identification that lets you connect to a datasource. Permissions within specific databases are controlled by user accounts. The database administrator maps your login account to a user account in any database you are authorized to access.

Logins are one of the fundamental security mechanisms of Performance Center. Users who connect to an Performance Center Server must identify themselves using a specific login identifier (ID). Users can then only see the tables and views they are authorized to see. To view Performance Center, a user need not login to the Performance Center Server. However, to add, modify, delete and access other functionality, a user **MUST** login to the target Performance Center Server **WITH** administrative privileges.

M**MDAC**

Microsoft Data Access Component.

Performance Center Server uses MDAC to establish a connection to the repository database and monitored datasources.

MTS

Multi-Threaded Server.

Oracle MTS configuration permits many user processes to share very few server processes. The user processes connect to a dispatcher background process, which routes client requests to the next available shared server process.

N**Notification**

Reusable messages that you tell Performance Center to send in response to threshold violations.

Notifier

The Performance Center Server internal process that is responsible for sending the emergency contact messages.

O**Oracle Alert Log**

An operating system file containing vital information on the Oracle database. The alert log contains information messages, including messages regarding the startup and shutdown of the database, errors that have occurred in the database. When a database is online Oracle continuously writes to the alert log.

P

Performance Category View

HTML windows where you can see statistics for each category of statistic. Every performance category view displays the Health Index and includes links to the Alert Log and Alarm views.

Pin

To hold fast; immobilize.

Marking an object to be pinned means Performance Center Server does not flush the pages for the object from memory. Pinning an object loads the object instance into the object cache, letting you access and modify the instance's attributes and follow references from that object to other objects.

Purge

To systematically and permanently remove old and unneeded data from an Performance Center Server. Purge is different from delete in that it is possible to regain deleted objects but purged objects are gone forever.

Q

There are no entries for this letter.

R

Repository

The Microsoft SQL Server, Oracle, DB2, or Sybase database server that Performance Center uses as a repository to store information used during monitoring.

Requestor

The Performance Center Server internal process that is responsible for collecting datasource data.

S

SID

System identifier.

A unique number that Oracle gives to a session.

Statistic

Individual performance measurements that Performance Center uses to gauge the overall health of the database.

T

Threshold

Default or user-defined values that mark performance boundaries. Thresholds are the point that must be exceeded to begin producing a user-defined effect or result or to elicit a response.

Alert Template

Logical containers used to define and deploy thresholds to monitored datasources.

U

There are no entries for this letter.

V

Views

HTML windows in Performance Center that are designed to provide you with several different ways to monitor your enterprise. The views include a combination of graphics and a color coded severity level, which let you quickly determine the health of all registered datasources in your enterprise.

W

WSH

Windows Scripting Host.

A Microsoft tool that the Performance Center Server uses to run Visual Basic scripts.

X

There are no entries for this letter.

Y

There are no entries for this letter.

Z

There are no entries for this letter.

Index

Symbols

% Privileged Time 310, 388, 479, 592
% User Time 310, 389, 479, 592

A

Action 634
Actions 148

- Adding 163
- Deleting 165
- Editing 163
- Maintenance 163
- Sample 73

Active Connections 244

- SQL Server 273
- Sybase 325, 368

Active Jobs 438
Active Jobs Tab 442
Active Rollbacks 439
Active Sessions 507

- Oracle 401

Active Transactions 273
Ad Hoc SQL Hit Ratio 238, 247
Additional resources 22
Address Lock Contention 358
Addresses 135

- Adding 136
- Deleting 140
- Editing 136
- Maintaining 136
- Types 136

Administrator Login 59
Agent Information 297
AIX 302, 381, 471, 584
Alarm Notifications 204
Alarms

- Acknowledging 203
- Getting 202

Alarms View 38, 44, 201, 634
Alert Log View 38, 204

- Finding Alerts 206
- Information Dialog Box 206

Alert Logs View 634
All Locks Tab

- Oracle 575
- SQL Server 279, 575
- Sybase 364, 576

All User Locks Tab 577
All Waits 504
Analyzer 634
Applications Tab 373
Archive Files Written Today 437
Archive View 38, 572, 634
Arrange Icons 71
Auto-Param Attempts 275
Available Paged Memory 306, 385, 475, 589
Available Physical Memory 306, 385, 475, 588
Available Swap Memory 307, 385, 476, 589

Average Disk Queue 312, 391, 595
Average Disk Queue Length 305, 383, 474, 587
Average User Database Volume 69

B

Backup/Restore Throughput 262
Backup/Restore T-Put 262
Backups Tab 296
Blackout 143, 165, 170, 634
Blackout Tab 172
Blocked 363
Blocked Users

- SQL Server 281
- Sybase 352

Blocking Lock Detail 287
Blocking Lock Rate 321
Blocking Lock Summary 287
Blocking Locks Tab

- Oracle 578
- SQL Server 279, 286, 579
- Sybase 365, 375, 580

Blocks Input/Sec 313, 391, 595
Blocks Output/Sec 313, 391, 482, 595
Bottleneck Analysis

- OS 303, 585

Buffer Busy Wait Ratio 489
Buffer Busy Waits Tab 497
Buffer Cache 249
Buffer Cache Hit Ratio

- SQL Server 237, 246

Buffer Cache Tab 251
Buffer Cache Usage 250
Buffer Pool Allocations 423
Buffer Pool Objects Detail 466
Buffer Pool Objects Summary 466
Buffer Pool Tab 466
Buffer Pool Usage 422
Buffer Size 315, 393, 484, 597
Bulk Copy Rows 262
Bulk Copy Throughput 262
Bytes from Client 403
Bytes from DBLink 404
Bytes per I/O Operation 312, 391, 595
Bytes Reads 263
Bytes Received 328
Bytes Received from Client 503
Bytes Received from DBLink 504
Bytes Sent 328
Bytes Sent to Client 503
Bytes Sent to DBLink 503
Bytes to Client 403
Bytes to DBLink 404
Bytes Written 263

C

Cache Activity Tab 336
Cache Allocations 335

- Cache Bindings
 - Index Statistics 355
 - Storage 356
 - Table Statistics 354
- Cache Detail Tab 348
- Cache Efficiency 315, 393, 484, 597
- Cache Size 315, 393, 484, 597
- Capture Agent 634
- Category 635
- Chained Table Placement 459
- Chained Tables Tab 467
- Clean Buffer Grab Rate 321
- Clock Algorithm Scans 315, 393, 484, 597
- Clone 635
- Clustered Table Inserts 377
- Committed Transactions 326, 369
- Completed Disk I/Os 338
- Configuration View 207
- Consistent Reads 442
- Contention
 - Number of Deadlocks 282
- Contention Detail View
 - Oracle 497
 - SQL Server 284
 - Sybase 364
- Contention Health Index 573
- Contention Page Statistics DB2
 - Applications Waiting on Locks 549
 - Average Lock Wait Time 550
 - Deadlocks 547
 - Lock Escalations 548
 - Lock List Utilization 549
 - Lock Overview 549
 - Lock Timeouts 548
 - Lock Wait Time 550
 - Lock Waits 548
- Contention Performance Category View 37, 43, 48
 - Statistics 356, 486
- Context Switches/Sec 310, 389, 480, 593
- Copy Read Hits % 315, 393, 484, 597
- CPU Busy 284
- CPU Events 310, 389, 592
- CPU Tab 309, 388, 478, 591
- CPU Utilization 309, 388, 479, 592
- Current Locks
 - Oracle 401, 507
 - SQL Server 244
 - Sybase 326, 361, 368
- Current Transactions 507
- D
- Data Cache Allocations 335
- Data Cache Hit Activity 329
- Data Cache Hit Rate 318
- Data Map Hits % 315, 394, 484, 597
- Database Detail 294
- Database Detail View
 - Sybase 352
 - Table Details 298
- Database Downtime Detail 69
- Database Home Page 288
- Database I/O Tab 261
- Database Low on Space 326
- Database Overview
 - SQL Server 265
 - Sybase 345
- Database Performance Comparison 69
- Database Reads 263
- Database Summary
 - SQL Server 293
 - Sybase 351
- Database Writes 263
- Databases 289
- Databases Low on Space 242
- Databases Needing Backup 291
- Databases Performance Category View 42, 47
 - Statistics 351
- Databases Without Auto-Create Stats 293
- Databases Without Auto-Grow 291
- Databases Without Auto-Update Statistics 292
- Datafile Fragmentation 455
- Datasource
 - Disabling 108
 - Disabling, Manually 108
 - Enabling 108
 - Enabling, Manually 108
 - Managing 105
 - Properties 109
 - Purging 107
 - Registering 109
- Datasource Health Index 573
 - Defaults 61
 - Properties 61
- Datasource Manager 635
- Datasource Menu 60
- Datasource Recorded Statistics 66
- Datasource Recorded Statistics Dialog Box 186–187
- Datasource Thresholds 180
 - Adding 182
 - Deleting 183
 - Editing 182
 - Maintaining 60
- Datasources
 - Selecting 105
- DB2 Diagnostic Log View 204
- DB2 Expert Guide 523
- DBArtisan 33
- DBCC Logical Scans 261
- DBWR 635
- Deadlock Rate 322, 360
- Deadlocks 241
 - Sybase 361
- Deferred Deletes 377
- Deferred Updates 377
- Delayed Disk I/Os 339–340
- Delete
 - Export Job 200
- Detach Blackout Schedule 171
- Device Detail 316, 395, 486, 599
- Device Detail Tab 348
- Device I/O Contention 323, 358
- Device Overview 347
- Device Summary 316, 395, 486, 599

Devices Tab 343

Dialog Box

- Action Maintenance 148
- Action Properties 164
- Add Statistic to Chart 213
- Address Detail 139, 144
- Address Maintenance 135
- Address Properties 139, 144
- Administrator Login 59
- Advanced Schedule Options 168–169
- Alarm Acknowledgement 204
- Alarm Notification 203
- Alarms Criteria 202
- Alert Log Information 206
- Available Blackout Schedules 169
- Blackout Tab 169
- Change Group 122
- Clone Template 191
- Datasource Alert Maintenance 127
- Datasource Health Index Maintenance
 - Add 131
 - Clone 131
 - Edit 131
- Datasource Health Index Properties 131
- Datasource Recorded Statistic 185
- Datasource Recorded Statistics 186
- Datasource Threshold Maintenance 180
- Datasources 109
- Delete Group 123
- Editing 32
- Find 206, 211
- Health Index Properties 128
- Health Index Templates 128
- Health Index Weight 134
- Maintenance 72
- Manage Datasources 106
- New Group 121
- New Profile 124
- Notification Maintenance 141
- Notification Properties 139, 143
- Recorded Statistic Template Maintenance 185
- Recorded Statistic Template Properties 189–190
- Remove Profile 125
- Rename Group 121
- Rename Profile 124
- Report 192
- Report Maintenance 193
- Save As 53
- Schedule Maintenance 165
- Schedule Properties 167–168
- Select Datasource 105
- Select Datasources 192
- Select Historical Report 193
- Select Notification Schedule 143, 170
- Select Performance Center Server 57
- Select Recipients 139
- Select Schedule 190
- Statistic Properties 128
- Statistics Maintenance 125
- Switch Profile 125
- Sybase Error Log Information 207
- Threshold Properties 167, 169, 182

- Threshold Template Maintenance 184
- Threshold Template Properties 188
- Threshold Templates Properties 188
- Direct Cheap Updates 377
- Direct Deletes 377
- Direct Expensive Updates 377
- Direct in Place Updates 377
- Dirty Buffers Grabbed 330
- Dirty Read Requests 330
- Dirty Read Restarts 331
- Discussion Groups 22
- Disk Analysis
 - OS 303, 585
- Disk Free Space 265
- Disk I/O Structures 359
- Disk I/O Time 312, 391, 595
- Disk Sort Detail Tab 513
- Disk Space Tab 270
- Disk Summary By Database 271
- Disk Summary By Space 271
- Disk Time 304, 382, 473, 586
- Disk Transfers/Sec 312, 391, 595
- Display Settings Dialog Box 227
- DML Detail Tab 377
- Documentation 22

E

Editing Result Sets

- Columns 226
- Copying 225
- Cutting 225
- Formatting Result Sets 227
- Pasting 225
- Resizing Columns 227
- Resizing Rows 226
- Rows 225
- Sorting Data 227

Editing SQL Scripts 220

- Copying and Pasting a Column 221
- Copying and Pasting Text 221
- Cutting and Pasting Text 221
- Selecting a Block of Text 220
- Selecting Text 220
- Setting Bookmarks 221–222

Embarcadero Performance Center Message Viewer 53

Embarcadero SQL Tuner 209

Embarcadero_Event Slots 145

Engine Configuration Limit 360

Engines Tab 344

Enqueue Waits 398, 496

Enterprise 635

Enterprise Connection Database Volume 69

Enterprise Database Availability 69

Enterprise Database Performance 69

Enterprise Database Performance by Category 69

Enterprise View 29, 635

- Web Client 30

Errors in Current Log 289

Event Log 53, 138

Explain Plan 211

Explorer 54

Export Data 195

- Export Job
 - Delete 200
 - Run Now 200
- Extent Deallocations 261
- Extents Allocated 261
- Extents Deficits Tab 456
- F
- Failed Auto-Param Attempts 275
- FAQs 22
- File Group Summary 267
- File Groups 290
- File Groups/Files Tab 267
- File I/O Tab 263
- File Summary 267
- Files 289
- Format Styles Dialog Box 228
- Formatting Result Sets 227
 - Changing Display Properties 229
 - Display Properties 229
 - Setting Alignment Properties 228
 - Setting Border Properties 229
 - Setting Colors 230
 - Setting Fonts 229
- Forums 22
- Forwarded Record Fetches 259
- Forwarded Records 262
- Fragmentation Tab
 - SQL Server 269
- Free Disk Space 308, 387, 477, 590
- Free List Waits 496
- Free Memory in Shared Pool 424
- Free Space 402
- Freespace Scans 261
- Full Scans 257, 262
- Full Screen Mode 71
- G
- Glossary 634
- Group Commit Sleeps 358
- H
- Health Index 207
- Health Index Properties 128
- Health Index Template 636
 - calculate 85
 - editing 85
- Health Index Templates 130
 - Adding 131
 - Applying 135
 - Cloning 131
 - Deleting 135
 - Editing 131
- Health Index View 38, 44, 49, 573
- Health Indexes 128
- Heap Table Inserts 377
- Heath Index 636
- Help Menu 71
- HIDD_OS_PC_TOPMEMPROC_ORA 590
- HIDD_SYB_AVG_BYTES_RECEIVED 328
- High Watermarks Tab 468
- Home View 636
- Home View Statistics 236
 - DB2
 - Hash Join Overflow Percentage 527
 - Sort Overflow Percentage 527
 - Oracle
 - Contention Vital Signs 397
 - I/O Vital Signs 399
 - Memory Vital Signs 397
 - Network Vital Signs 403
 - Space Vital Signs 402
 - Users Vital Signs 400
 - SQL Server
 - Contention Vital Signs 240
 - I/O Vital Signs 239
 - Memory Vital Signs 237
 - Network Vital Signs 245
 - Space Vital Signs 242
 - Users Vital Signs 243
 - Sybase 318
 - Contention Vital Signs 321
 - I/O Vital Signs 324
 - Memory Vital Signs 318
 - Network Vital Signs 327
 - Space Vital Signs 326
 - Users Vital Signs 325
- Hot Objects View 38, 573
 - Hot Code Tab 574
 - Hot Tables Tab 573
- HP-UX 302, 381, 471, 584
- I
- I/O 636
- I/O Busy 284
- I/O Detail 260
- I/O Detail View
 - Sybase 342
- I/O Details 312, 391, 481, 594
- I/O Error Rate
 - Sybase 340
- I/O Errors 240, 260, 324
- I/O Home Page 253

- I/O Page Statistics
 - Asynchronous Read Rate 534
 - Asynchronous Write Rate 535
 - Asynchronous Write Ratio 533
 - Buffer Pool Hit Ratio 536
 - Buffer Pool Index Hit Ratio 537
 - Direct Read Rate 534
 - Direct Write Rate 534
 - Direct Write Ratio 533
 - Dirty Page Cleans 536
 - Log Read Rate 534
 - Log Space Cleans 535
 - Log Space Cleans Ratio 537
 - Log Write Rate 534
 - Page Cleaners 535
 - Prefetch Wait Time 536
 - Prefetchers 535
 - Session Leaders – I/O 538
 - Synchronous Read Rate 534
 - Synchronous Write Rate 535
 - Synchronous Write Ratio 533
 - Threshold Cleans Ratio 537
 - Victim Cleans Hit Ratio 537
- I/O Performance Category View 36, 41, 47
 - Statistics 338, 531
- I/O Stall 263
- I/O Tab 626
- Inactive Connections
 - SQL Server 273
 - Sybase 368
- Inactive Sessions 508
- Inbound Packet Errors 395, 486, 599
- Inbound Packets Discarded 317, 395, 486, 599
- Index (ROWID) Accesses 460
- Index Demographics 354
- Index Details Tab 297
- Index Reserved 267
- Index Scans 438
- Index Scans Tab 345
- Index Searches 258, 262
- Index Statistics Tab 354
- Indexes Tab 269
- INET Address 317, 395, 486, 599
- Init.ora 636
- Instance Page Statistics DB2
 - Agent Utilization 571
 - Assigned from Pool 568
 - Buffer Utilization 565
 - Connection Entry Utilization 565
 - Created Empty Pool 568
 - Database Configuration 569
 - FCM Resource Utilization 571
 - FCM Throughput 570
 - Idle 567
 - Instance Identification 566
 - Instance Utilities 570
 - Memory Pool Size 570
 - Memory Pool Utilization 569
 - Memory Pools 569
 - Message Anchor Utilization 565
 - Monitor Heap Utilization 562
 - Monitor Switches 566
 - Percent of Agents Stolen 571
 - Piped Sorts Rejected 563
 - Piped Sorts Requested 563
 - Post Threshold Hash Joins 564
 - Post Threshold Sorts 564
 - Private Memory (MB) 563
 - Registered 567
 - Request Block Utilization 565
 - Request Overflows 568
 - Requests 567
 - Sort Heap (MB) 562
 - Sort Heap Utilization 562
 - Stolen 567
 - Waiting for Tokens 568
- Instance Parameters View 39
- Integrated Security 95, 113–114
- Interrupts/Sec 310, 389, 479, 592
- Invalid/Unusable Objects 463
- Invalid/Unusable Objects Detail 465
- Invalid/Unusable Objects Summary 465
- ISQL Window 214
 - Bookmarks 221
 - Clearing 222
 - Goto 222
 - Moving 222
 - Editing Result Sets 225
 - Executing Scripts 223
 - Cancelling a Query 223
- ISQL Windows 214
 - Find and Replace 215
 - Replacing 217
 - Searching 216
- K
- Key Resource Usage
 - OS 302, 585
- Knowledge Base 22
- L
- Large I/O Acquired 334
- Large I/O Denied 334
- Large I/O Hit Rate 320, 333
- Last Full Backup 294
- Latch Detail Tab 497

- Latch Immediate Miss Ratio 488
 - Latch Miss Ratio 487
 - Latch Sleep Ratio 488
 - Latch Waits 241
 - SQL Server 282
 - Latch Waits Tab 498
 - Leading Sessions
 - Oracle
 - CPU 511
 - I/O 512
 - Memory 512
 - Leading Sessions Tab
 - Oracle
 - I/O 443
 - Memory 433
 - SQL Server
 - I/O 264
 - Memory 253
 - Sybase
 - I/O 342
 - Memory 335
 - LGWR 636
 - Linux 302, 381, 471, 584
 - Load Average 304, 383, 473, 586
 - Load Averages 311, 390, 593
 - Local Filesystem 316, 485
 - Lock Contention
 - Sybase 362
 - Lock Detail 286
 - Lock Overview 283
 - Lock Summary 285
 - Lock Timeouts 241
 - Lock Waits 492
 - Locked Objects 464
 - Locks 207
 - Locks Tab
 - SQL Server 285
 - Sybase 374
 - Locks View 39, 45, 49, 574
 - All Locks Tab 575
 - All User Locks Tab 577
 - Blocking Locks Tab 578
 - Locks View for DB2 581
 - Log Cache Hit Ratio 238
 - Log Cache Reads 252, 262
 - Log Cache Tab 252
 - Log Flush Wait Time 252
 - Log Flush Waits 252, 283
 - Log Flushes 239, 252, 262
 - Log Growths 252, 262
 - Log Overview
 - SQL Server 266
 - Sybase 346
 - Log Shrinks 252, 262
 - Log Truncation 262
 - Logical Changes 400, 442
 - Logical I/O 441
 - Logical Lock Contention 357
 - Logical Reads 400, 442
 - Login 637
 - Login Wizard
 - Panel 1 for Microsoft SQL Server 197
 - Panel 1 for Sybase ASE 199
 - Panel 2 for Microsoft SQL Server 197
 - Panel 2 for Sybase ASE 199
 - Panel 3 for Microsoft SQL Server 197
 - Panel 3 for Sybase ASE 199
 - Panel 4 for Sybase ASE 200
 - Login, Administrator 59
 - Logins 245
 - Logouts 245
 - Logs Low on Space 243
 - Sybase 327
 - Logs Without Auto-Grow 292
 - Long Table Scan Rows 460
 - Long Table Scans 438
- M
- Maintenance Dialog Boxes 72
 - Actions 73
 - Addresses 73
 - Health Index Templates 84
 - Notifications 73
 - Recorded Statistic Templates 90
 - Reports 76
 - Statistics 75
 - Threshold Templates 76
 - Maintenance Menu 68
 - Manage with DBArtisan 33
 - Max Open Locks 507
 - Max Query Slaves 495
 - Max Sessions 508
 - Max Transactions 507
 - MDAC 637
 - MDL Read Hits % 315, 394, 484, 597
 - Memory Analysis
 - OS 303, 585
 - Memory Available 315, 393, 484, 597
 - Memory Detail 251
 - Memory Detail View
 - Sybase 335
 - Memory Freed 315, 393, 484, 597
 - Memory Health Index 573
 - Memory Home Page 246
 - Memory Page Statistics
 - Shared Workspace Hit Ratio 531
 - Memory Page Statistics DB2
 - Catalog Cache Overflows 528
 - Lock List Utilization 529
 - Package Cache Hit Ratio 530
 - Package Cache Overflows 528
 - Private Workspace Overflows 528
 - Session Leaders – Memory 530
 - Shared Workspace Overflows 529
 - Sort Heap Utilization 530
 - Memory Performance Category View 36, 41, 46
 - Statistics 328
 - Memory Tab 312, 391, 482, 595, 623
 - Memory Usage 248

- Menu 56
 - Datasource 60
 - Help 71
 - Maintenance 68
 - Monitor 57
 - Report 68
 - View 70
 - Window 71
- Message Viewer 53
- Migrate Data 195
- Modify Conflicts 358
- Monitor Menu 57
- Mounted On 316, 486
- MTS 637
- MTS Request Wait Time 505
- MTS Request Waits 505
- MTS Response Activity 505
- MTS Response Waits 505
- MTS Response Waits Time 505
- Multiple Statistic Baseline Analysis 69
- Multiple Statistic Trend Analysis 69
- N
- Network Bytes Read 301
- Network Bytes Written 301
- Network Contention 504
- Network Contention Rate 323, 363, 380
- Network Delays 328, 364, 380
- Network Details 316, 395, 486, 599
- Network Errors 379
- Network I/O Wait Requests 301
- Network I/O Wait Time 302
- Network Performance Category View 38, 44, 48
 - Statistics 299, 378, 502
- Network Queue Length 306, 384, 475, 588
- Network Reads 300
- Network Requests 328, 364, 381
- Network Waits 504
- Network Writes 300
- New Pages Allocated 330
- Notification 637
- Notifications 141
 - Adding 142
 - Attaching 147
 - Deleting 147
 - Editing 142
 - Maintaining 142
- Notifier 637
- Number of Logins 309, 388, 478, 591
- Number of Processes 309, 388, 478, 591
- O
- Object Detail Tab 350
- Object Extents Tab 208, 469
- Object Page Statistics DB2
 - Buffer Pools Tab – I/O Analysis 543
 - Buffer Pools Tab – Overview 543
 - Containers Tab 546
 - Tables Tab 546
 - Tablespaces Tab – I/O Analysis 545
 - Tablespaces Tab – Overview 544
- Object Summary Tab 452
- Objects
 - Extent Problems 462
 - Objects Accessed 521, 601
 - Objects Accessed Tab 468
 - Objects Health Index 573
 - Objects in Memory Tab 434, 470
 - Objects Performance Category View 37
 - Objects Pinned 463
 - Offline Databases 291
 - Open Cursor Detail 515
 - Open Cursor Summary 515
 - Open Cursors 509
 - Open Cursors Tab 515
 - Operating System
 - Monitoring 114
 - Operating System Limit 360
 - Operating System Statistics 302, 381, 471, 584
 - Operating System View 39, 45, 49, 583
 - Options Editor
 - Client 90
 - Category Options 93
 - Datasource Options 93
 - General Options 91
 - Monitor Server Options 94
 - SQL Editor Options 93
 - Statistic Options 92
 - Server 94
 - Analyzer Options 96
 - Capture Agent Options 96
 - Datasource Manager Options 98
 - Notifier Options 99
 - Report Options 102
 - Repository Options 97
 - Requestor Options 102
 - Run As Options 95
 - Options Tab
 - Sybase 356
- Oracle Alert Log 637
- Oracle Buffer Wait Statistics 69
- Oracle Buffer Wait Statistics Comparison to Baseline 69
- Oracle Instance Efficiency Ratios 69
- Oracle Instance Efficiency Ratios Comparison to Baseline 69
- Oracle Load Profile 69
- Oracle Load Profile Comparison to Baseline 69
- Oracle Statspack 195
- Oracle Top Wait Events 69
- Oracle Top Wait Events Comparison to Baseline 69
- OS Detail 309, 388, 591
- OS Drill Down 309, 388, 591
- OS Statistics 302, 381, 471, 584
- OS View
 - Processor 304, 382, 473, 586
 - Processor Speed 303, 382, 472, 586
- Outbound Packet Errors 317, 395, 486, 599
- Outbound Packets Discarded 317, 395, 486, 599
- Output Window 51
 - Notifications Tab 52
 - Status Tab 52
 - Threshold Events Tab 52
- Outstanding Disk I/Os 339

Overview Tab 294

P

Packet Collisions 317, 395, 486, 599
 Packet Discard 317, 395, 486, 599
 Packet Discards 317, 395, 486, 599
 Packet Errors 301, 317, 395, 486, 599
 Packets Received 245, 300
 Packets Received/Sec 317, 395, 486, 599
 Packets Sent 245, 300
 Packets Sent/Sec 317, 395, 486, 599
 Page Activity Tab 337
 Page Deadlocks 281
 Page Faults/Sec 305, 384, 474, 587
 Page Reads 254
 Page Replacements 315, 393, 484, 597
 Page Splits 261
 Paged In 313, 391, 482, 595
 Paged Memory Used 304, 383, 473, 586
 Paged Out 313, 391, 482, 595
 Pager 138
 Pages Input/Sec 313, 391, 482, 595
 Pages Output/Sec 313, 392, 482, 595
 Paging Activity 313, 391, 482, 595
 Parallel Query Busy Ratio 494
 Parallel Query Tab 499
 Partition 316, 395, 485, 598
 Performance Analyst for Microsoft Statistics 236
 Database Page
 Database Detail
 Backups Tab 296
 SQL Agent Tab
 SQL Server Alert Summary 296
 SQL Server Job Summary 295
 I/O Page
 Total I/O 260
 Memory Page
 Memory Detail 251
 Users Page
 Active Transactions 273
 Performance Analyst for Oracle
 Statistics
 Memory Page
 Memory Drill Down 430
 Performance Analyst for Oracle Expert Guide 396

Performance Analyst for Oracle Statistics 396

Home Page

Active User Processes 416
 Archive Log 416
 Buffer Busy Waits 409
 Current Object Blocks 414
 Dictionary Cache Hit Ratio 407
 Enqueue Waits 415
 Free List Waits 415
 Free Shared Pool Percent 411
 Inactive User Processes 417
 Latch Miss Ratio 410
 Library Cache Hit Ratio 405
 Memory Sort Ratio 408
 Parallel Query Busy Ratio 411
 Parse/Execute Ratio 409
 Problem Objects 413
 Problem Tablespaces 412
 Rollback Contention Ratio 410
 Top Session Bottlenecks 414
 Top System Bottlenecks 413
 Total Free Space 415
 Total Used Space 415

I/O Page

I/O Detail 442
 Datafile I/O Tab 444
 DBWR/LGWR Tab 447
 Database Writer Detail 447
 Redo Wastage 447
 Rollback I/O Tab 445
 Rollback I/O 445
 I/O Drill Down 442

Memory Page 417

Bottleneck Analysis 418
 Buffer Cache Hit Ratio 421
 Dictionary Cache Hit Ratio 426
 Dictionary Cache Tab 431
 Key Ratio Analysis 418
 Library Cache Hit Ratio 425
 Memory Detail 430
 Library Cache Tab 432
 Shared Pool Tab 436
 Memory Sort Ratio 427
 SGA Analysis 420
 SQL Analysis 419
 Top Memory Hogs 421
 Workload Analysis 421

Objects Page 457

Invalid Objects 463
 Locked Objects 464
 Object Buffer Pool Placement 458
 Objects Detail 464
 Invalid Objects Tab 465
 Objects Drill Down 464
 User Object Analysis

- Active Rollback Ratio 459
- OS Page 302, 381, 471, 584
 - Available Paged Memory 306, 385, 475, 589
 - Available Physical Memory 306, 385, 475, 588
 - Available Swap Memory 307, 385, 476, 589
 - Average Disk Queue Length 305, 383, 474, 587
 - Disk Time 304, 382, 473, 586
 - Free Disk Space 308, 387, 477, 590
 - Load Average 304, 383, 473, 586
 - Network Output Queue Length 306, 384, 475, 588
 - Network Queue Length 306, 384, 475, 588
 - Number of Logins 309, 388, 478, 591
 - Number of Processes 309, 388, 478, 591
 - OS Detail 309
 - CPU Tab 309, 388, 478, 591
 - CPU Events 310
 - CPU Events Context Switches/Sec 310, 389, 480, 593
 - CPU Events Interrupts/Sec 310, 389, 479, 592
 - CPU Events System Calls/Sec 311
 - CPU Utilization 309, 388, 479, 592
 - CPU Utilization % Privileged Time 310, 388, 479, 592
 - CPU Utilization % User Time 310, 389, 479, 592
 - Load Averages 311
 - Processor Queue Length 311, 389, 480, 593
 - I/O Tab 312, 391, 481, 594
 - Average Disk Queue 312
 - Bytes per I/O Operation 312
 - Disk I/O Time 312
 - Total Disk Queue 312
 - Transfer Rate 312
 - Memory Tab 312, 391, 482, 595
 - Buffer Size 315, 484
 - Cache Efficiency 315, 393, 484, 597
 - Cache Efficiency Copy Read Hits % 315, 393, 484, 597
 - Cache Efficiency Data Map Hits % 315, 394, 484, 597
 - Cache Efficiency MDL Read Hits % 315, 394, 484, 597
 - Cache Efficiency Pin Read Hits % 316, 394, 485, 598
 - Cache Size 315, 484
 - Memory Available 315, 484
 - Page Replacements 315, 484
 - Page Replacements Clock Algorithm Scans (Pages/sec) 315, 484
 - Page Replacements Memory Freed (Pages/sec) 315, 484
 - Paging Activity 313, 391, 482, 595
 - Paging Activity Blocks Output/Sec 313, 482
 - Paging Activity Paged In 313, 482
 - Paging Activity Paged Out 313, 482
 - Paging Activity Pages Input/Sec 313, 391, 482, 595
 - Paging Activity Pages Output/Sec 313, 392, 482, 595
- Network Tab 316, 395, 486, 599
 - Inbound Packet Errors 317, 486
 - Network Details 316, 486
 - Outbound Packet Errors 317, 486
 - Packet Collisions 317, 486
 - Packet Discards 317, 486
 - Packet Discards Inbound Packets Discarded 317, 486
 - Packet Discards Outbound Packets Discarded 317, 486
 - Packet Errors 317, 486
 - Transmission Queue Length 317, 486
 - Transmission Rate 317, 486
 - Transmission Rate (Bytes) 317, 486
 - Transmission Rate Packets Received/Sec 317, 486
 - Transmission Rate Packets Sent/Sec 317, 486
 - Transmission Received Rate (KB/Sec) 317, 486
 - Transmission Sent Rate (KB/Sec) 317, 486
- Processes Tab 311, 390, 480, 593
- Space Tab 316, 394, 485, 598
 - Device Detail 316, 486
 - Device Summary 316, 486
- Page Faults/Sec 305, 384, 474, 587
- Paged Memory Used 304, 383, 473, 586
- Processor Queue Length 305, 384, 474, 588
- Processor Time 303, 382, 472, 585
- Swap Memory Used 304, 383, 473, 587
- Top CPU Process 309, 387, 478, 591
- Top I/O Process 309, 388, 478, 591
- Top Memory Process 308, 387, 477, 590
- Total Disk Space 308, 386, 477, 590
- Total Paged Memory 307, 386, 476, 589
- Total Physical Memory 307, 386, 476, 589
- Total Swap Memory 307, 386, 476, 589
- Used Disk Space 307, 386, 476, 590
- Session Details Page 519
- Space Page
 - Space Detail 450
 - Space Fragmentation Tab 454
 - Tablespace Fragmentation 454
 - Tablespace Growth Tab 453
 - Space Drill Down 450
- Statistics
 - Memory Page
 - Free Shared Pool Percent 428
- Users Page 506
 - Users Detail 513
 - Users Drill Down 513
- Performance Analyst Microsoft SQL Server Expert Guide 236
- Performance Category View 638
- Performance Category Views 31
 - Customizing 32
 - Menu 32
 - Navigating 32
 - Shortcut Menu 33

- Performance Center Server Manager 70
- Physical I/O 441
- Physical Reads 399, 441
- Physical Writes 400, 441
- Pin 638
- Pin Read Hits % 316, 394, 485, 598
- Pinned Objects 251
- Pinned Tables 297
- Postmsg Format 145
- Probe Scans 258, 262
- Procedure Cache Activity 331
- Procedure Cache Allocations 335
- Procedure Cache Hit Rate 319
- Procedure Hit Ratio 247
- Procedure Plan Hit Ratio 237
- Procedure Reads from Disk 332
- Procedure Removals 333
- Procedure Requests 332
- Procedure Writes to Disk 333
- Processes Tab 311, 390, 480, 593
 - SQL Server 278
 - Sybase 374
- Processor Queue Length 311, 389, 480, 593
- Processor Time 303, 382, 472, 585
- Purge 638
- Q
- Query Plan 211
- R
- Range Scans 258, 262
- Received (KB/Sec) 317, 395, 486, 599
- Recorded Statistic Templates 185
 - Adding 187
 - Applying 191
 - Cloning 191
 - Deleting 190
 - Editing 187
 - Maintaining 186
 - Properties 189
- Recorded Statistics
 - Datasource 66
- Redo Log Size 440
- Redo Log Space Requests 492
- Redo Log Space Wait Time 398, 493
- Redo Wastage 440
- Redo Writes 441
- Replication Tab 296
- Report Maintenance 192
- Report Menu 68
- Reports 192
 - Average User Database Volume 69
 - Database Downtime Detail 69
 - Database Performance Comparison 69
 - Enterprise Connection Database Volume 69
 - Enterprise Database Availability 69
 - Enterprise Database Performance 69
 - Enterprise Database Performance by Category 69
 - Generating 192
 - Multiple Statistic Baseline Analysis 69
 - Multiple Statistic Trend Analysis 69
 - Oracle Instance Efficiency Ratios 69
 - Oracle Instance Efficiency Ratios Comparison to Baseline 69
 - Oracle Load Profile 69
 - Oracle Load Profile Comparison to Baseline 69
 - Oracle Top Wait Events 69
 - Oracle Top Wait Events Comparison to Baseline 69
 - Selecting 193
 - SGA Regions and Memory Areas 69
 - Statistical Trend Analysis 70
 - Worst Databases 70
- Repository 638
- Requested Disk I/Os 338
- Requestor 638
- Result Sets
 - Closing 224
 - Editing 225
 - Formatting 227
 - Saving 224
 - Saving as XML 224
- Results Editor 223
 - Closing a Result Set 224
 - Editing Result Sets 225
 - Opening Result Files 215
 - Renaming a Result Window 224
 - Saving Results 224
 - Status Bar 223
- Rollback Activity Detail 445
- Rollback Contention Ratio 489
- Rollback Gets 440
- Rollback Segments Accessed 521, 601
- Rollback Summary 464
- Rollback Waits 440
- Rollback Waits Tab 500
- Rollback/Optimal Detail 446
- Roundtrips to/from Client 503
- Roundtrips to/from DBLink 504
- Rows Deleted 371
- Rows Inserted 370
- Rows Updated 370
- Rows Updated (Data Locks Only) 371
- Run Now
 - Export Job 200
- S
- Sample Actions 73
- Schedules 165
- Schema Leaders - Space 450
- Script Files 215
- Segment Detail 355
- Segment Detail Tab 349

- Select Schedule Dialog Box
 - Opening 190
- Sent (KB/Sec) 317, 395, 486, 599
- Server
 - Pinging 58
 - Selecting 57
 - Starting 58
 - Stopping 58
- Server Configuration Limit 359
- Server Configuration View 207
- Server I/O Busy Rate
 - SQL Server 259
 - Sybase 339
- Server Idle 284
- Server Manager 70
- Session Contention Tab 520, 600
- Session Detail View 208
 - Oracle 519, 599
 - SQL Server 603
 - Sybase 609
- Session I/O Tab 519, 600
- Session Leaders
 - Oracle
 - I/O 439
 - Memory 429
 - SQL Server
 - CPU 277
 - I/O 259, 277
 - Memory 249, 276
 - Sybase
 - CPU 372
 - I/O 340, 372
 - Memory 334, 371
- Session Memory Tab 519, 600
- Session Network Tab 521, 602
- Session Objects Tab
 - Oracle 521, 601
- Session SQL Tab 521, 602
- Session Statistics Tab 522, 602
- Session Waits Tab 500, 516
- Sessions Blocked 490, 510
- Sessions in Buffer Busy Waits 491
- Sessions Involved in Disk Sorts 510
- Sessions Overview Tab 435, 517
- Sessions Waiting 509
- Sessions Waiting (General) 491
- Sessions Waiting for Latches 491
- SGA 430
 - Database Buffers 430
 - Fixed Size 430
 - Redo Buffers 430
 - Variable Size 430
- SGA Regions and Memory Areas 69
- Short Table Scans 438
- Shortcut Menus 72
- SID 638
- Skipped Ghosted Records 262
- SMS 138
- SNMP 138
- Solaris 302, 381, 471, 584
- Space Detail 267
- Space Detail View
 - Sybase 348
- Space Health Index 573
- Space Page Statistics DB2
 - Abnormal State Tablespaces 542
 - Active Log Free 540
 - Active Log Size 540
 - Active Log Used 540
 - Active Log Utilization 540
 - DMS Free 539
 - DMS Used 539
 - DMS Utilization 539
 - Inaccessible Containers 541
 - Secondary Log Used HWM 541
 - Secondary Logs Allocated 540
 - SMS Free 539
 - SMS Used 539
 - SMS Utilization 540
 - Tablespace Low on Space 541
 - Tablespace Overview 542
- Space Performance Category View 36, 42, 47
 - Statistics 345, 538
- Space Tab 316, 394, 485, 598
- Space Usage Tab 457
- SQL Agent Tab 295
- SQL Cache 248
- SQL Cache Details 251, 297
- SQL Cache Summary 251
- SQL Cache Tab 251
- SQL Compilations 274
- SQL Editor 214
 - Closing a Script 219
 - Closing the Error List 219
 - Editing a Script 220
 - Executing Scripts 223
 - Goto 218
 - ISQL Windows 214
 - Navigating 218
 - Opening ISQL Windows 214
 - Opening Script Files 215
 - Regular Expressions 217
 - Saving a Script 219
- SQL Re-Compilations 274
- SQL Scripting 214
 - Results Editor 223
 - SQL Editor 214
- SQL Server Login Wizard
 - Panel 1 118
 - Panel 2 118
 - Panel 3 118
- SQL Server Logs View 45, 204
- Statistic 638
- Statistical Trend Analysis 70
- Statistics 125
 - Maintenance 125–126
- Status Bar 53
- Storage Tab 355
- Suspect Databases 290, 352
- Suspect Objects 352
- Suspect Objects Tab 352
- Suspect Pages 352

- Swap Memory Used 304, 383, 473, 587
- Sybase Error Log View 204
 - Information Dialog Box 207
- Sybase Expert Guide 318
- Sybase Login Wizard
 - Panel 1 119
 - Panel 2 119
 - Panel 3 119
 - Panel 4 120
- System Calls/Sec 311, 389, 593
- System Connections 273
- System Tab 261
- System Tablespace Tab 517
- System Waits Tab 501
- T
- Table Demographics 353
- Table Details
 - Database Detail View 298
- Table Lock Escalations 283
- Table Reserved 267
- Table Statistics Tab 353
- Tables Tab 268
- Tablespace
 - Low on Space 402
- Tablespace Fragmentation 454
- Tablespace Low on Space 402
- Tablespace Map Tab 451
- Threshold 639
- Threshold Template 639
- Threshold Templates 184
 - Adding 187
 - Applying 191
 - Cloning 191
 - Deleting 190
 - Editing 187
 - Maintaining 186
 - Properties 188
- Tivoli
 - Examples of Events 145
 - Notifier Options 101
- Tivoli Configuration 143
- Tivoli eMsg 138
- Tivoli Notifications 141–143
- Toolbar 72
- Top CPU Process 309, 387, 478, 591
- Top I/O Process 309, 388, 478, 591
- Top Memory Process 308, 387, 477, 590
- Top Sessions 208
- Top Sessions View 39, 45, 49, 622
 - DB2
 - CPU Tab 632
 - I/O Tab 629
 - Memory Tab 625
 - Oracle
 - CPU Tab 631
 - I/O Tab 626
 - Memory Tab 623
 - SQL Server
 - CPU Tab 631
 - I/O Tab 627
 - Memory Tab 623
 - Sybase
 - CPU Tab 631
 - I/O Tab 628
 - Memory Tab 624
- Top SQL 209
- Top SQL View 39, 45
- Total Bytes Received 379
- Total Bytes Sent 379
- Total Connections
 - SQL Server 243, 272
 - Sybase 325
- Total Disk Queue 312, 391, 595
- Total Disk Space 308, 386, 477, 590
- Total Packets Received 379
- Total Packets Sent 379
- Total Paged Memory 307, 386, 476, 589
- Total Physical Memory 307, 386, 476, 589
- Total Rows Affected 371
- Total Server Reads
 - SQL Server 239
 - Sybase 324
- Total Server Writes
 - SQL Server 239
 - Sybase 324
- Total Sessions
 - Oracle 400, 508
- Total Swap Memory 307, 386, 476, 589
- Total Traffic 301
- Transaction Detail Tab 518
- Transaction Log Allocations 339
- Transaction Log Writes 324, 339
- Transactions 244
- Transactions Tab 377
- Transfer Rate 312, 391, 595
- Transmission Queue Length 317, 395, 486, 599
- Transmission Rate
 - Bytes 317, 395, 486, 599
 - Packets 317, 395, 486, 599
- Trends View 39, 45, 212
 - Adding a Statistic 213
 - Deleting a Statistic 213
- T-SQL Batches 273
- Tune SQL 209
- Txn Log Writes 324
- U
- ULC Flushes 370
- Unblackout 143, 171
- Unix 302, 381, 471, 584

- Used Disk Space 307, 386, 476, 590
- Used Memory in Shared Pool 424
- Used Query Slaves 494
- Used Space 402
- User Commits 438
- User Home Page 271
- User I/O Tab 262
- User Object Accesses 438
- User Rollbacks 438
- User Wizard
 - Panel 1 for Oracle 198
 - Panel 2 for Oracle 198
 - Panel 3 for Oracle 198
 - Panel 4 for Oracle 199
- Users Blocked
 - Oracle 398
 - SQL Server 276
 - Sybase 369
- Users Detail View
 - SQL Server 277
 - Sybase 373
- Users Health Index 573
- Users Page Statistics DB2
 - Connections 551
 - Connections Active 552
 - Connections Idle 552
 - Connections Waiting 552
 - Dynamic SQL Statements 552
 - Lock Waits Per Transaction 553
 - Max. Connections 555
 - Percent Executing 555
 - Percent of Maximum 555
 - Rows Selected Per Transaction 553
 - Selects Per Transaction 553
 - Session Leaders – CPU 554
 - Session Leaders – I/O 554
 - Session Leaders – Memory 553
 - Sorts Per Transaction 553
 - Static SQL Statements 552
 - Transactions 553
 - Transactions Per Second 553
- Users Performance Category View 38, 43, 48
 - Statistics 367, 551
- Users Waiting 397

∨

- View Menu 70
- Views 29, 639
 - Contention 37, 43, 48
 - Databases 42, 47
 - Enterprise 29
 - I/O 36, 41, 47
 - Memory 36, 41, 46
 - Network 38, 44, 48
 - Object 37
 - Other 38, 44, 49
 - Space 36, 42, 47
 - Users 38, 43, 48
 - Web Client 49, 232
- Virtual Log Files Tab 268
- Vital Signs 34, 39, 45

W

Waits Tab 288

Web Client

 Enterprise View 30

Web Client Views 49, 232

Window Menu 71

Windows 302, 381, 471, 584

WINS 139

Workbook 51

Worktables Created 258

Worst Databases 70

WSH 639