Monitor Databases

On this page:

- Set the Time Range
- View Historical Performance Data on the Activity window
- Set the View on the Activity Window
- Example Workflow Drilling Down Through Views
- Indepth Analysis of SQL Statements
- View Historical %CPU Consumption
- ViewUp-to-the-Second Performance (Current window)
- Database Statistics
- Database Objects

Access the Database Platform Window

From the **Main Menu** window, click a database icon to open the database platform window for that particular database. You can also directly access the platform window when you link to AppDynamics for Databases from AppDynamics Pro.

	Setup Dashboard	Alerts Help	
Setup	Alerts		Multi-Instance Dashboard
Manage data collection Manage license keys		Configure notifications for Performance Metrics.	View the performance status of your monitored environment.
Your Monitored Databases			
	BMDB2	Monitoring 1 instance	s) of DB2
		Monitoring 9 instance	s) of Oracle
	MySQL	Monitoring 7 instance	s) of MySQL
Monitored Infrastructure			
	NetApp	Monitoring 5 NetApp (Version 2.8	Controller(s)
	NetApp E-Series	Monitoring 1 E-Series Version 2.8	Array(s)
	Server	Monitoring 13 Server(s Version 2.8	5)

The database platform window appears.



On the Activity window, you can double-click an Entity to see the SQL details for it, and if relevant, as is the case with the Oracle database, you can also see the Explain Plan details.

	Current Activity Statistic	cs SQL Objects	Host: 🗸 or	rders Reports 🗸	Custom 🗸
SQL ID: 47j049cqcp	58j		Get	(
SELECT OWNER, SE FROM SYS_DBA_SEC WHERE SEGMENT_OF AND SEGMENT_TYPF AND SEGMENT_TYPF AND SEGMENT_TYPF AND SEGMENT_TYPF AND SEGMENT_TYPF AND SEGMENT_TYPF AND SEGMENT_TYPF AND TABLESPACE_N	EGMENT_NAME, PARTITION_NAME, SEG SQL GS BJD = :B1 E <> 'ROLLBACK' E <> 'TYPE2 UNDO' E <> 'DEFERRED ROLLBACK' E <> 'DEFERRED ROLLBACK' E <> 'CACHE' E <> 'SPACE HEADER' E <> 'UNDEFINED' NAME NOT IN ('SYSAUX', 'SYSTEM')	LESPACE_NAME, TABLESPACE			
🞝 Explain Plan (Fro	m cache) Explain				R
Number of steps: 120 Explained for orders in s	schema SYS				
Referenced Objects	3			Object	
Tree Node	Operation	Object Type	Object Name	details	
1	VIEW	VIEW	SYS_DBA_SEGS		View Object
1	VIEW	VIEW	SYS_DBA_SEGS		View Object
8	TABLE ACCESS BY INDEX ROWID	TABLE	OBJ\$		View Object
8	TABLE ACCESS BY INDEX ROWID	TABLE	OBJ\$		View Object
9	INDEX RANGE SCAN	INDEX	I_OBJ4		View Object

Multiple Oracle Execution Plans

Multiple execution plans are presented in some cases such as if there are bind variables associated with a statement. Oracle examines the content of the bind variables and the associated table statistics and may determine that one execution plan may be more efficient than another depending on the value of the bind variable. When appropriate, multiple execution plans display on the SQL window where you can compare execution statistics and drill down to the explain plan SQL to determine why one explain plan is preferable.

SQL ID 7a0sah244x8g2 SQL ID Report View Raw Stats Execution Plan(s) Plan_Hash_Value Executions Elapsed Time Image: SQL Executed by: Machine 1001 500 SQL Executed by: Machine Program Elapsed Time Image: WTitimegu Image: Stress of the securition of times JDBC Thin Client Image: Stress of the securition of the securition of times Stress of times Stress of times	5 12:05 🛐 Go
Execution Plan(s) Plan_Hash_Value Executions Elapsed Time Image: Association Plan(s) Image: Association Plan(s) 1001 Image: Association Plan(s) Image: Association Plan(s) 1001 <th></th>	
SQL Executed by: Machine 500 w7timcgu Times JDBC Thin Client SKGROUPWIN-S7CNIM71MI SYSTEM sqlplus.exe	
SQL Executed by: Machine SQL Executed by: Machine SQL Execution Program Elapsed W7Itimcgu VTimes JDBC Thin Client SQL Executed by: VICTURE SYSTEM sqlplus.exe	00:02:
SQL Executed by: Machine execution Program Elapsed w7timcgu JDBC Thin Client SKGROUPWIN-S7CNIM71MI SYSTEM sqlplus.exe	00:00:
w7timcgu times JDBC Thin Client	Time
Click to discovery KGROUP/WIN-S7CNIM71MI SYSTEM sqlplus.exe	00:03:
what makes one explain plan more efficient than the	100
Other by drilling down into their SQL 11:30 11:35 11:45 11:50 11:55 12:00	

Set the Time Range

For the **Activity** and **Statistics** tabs, you can select the time period using the **Date** list and calendar settings at the top right of the window.

Date: 🔻	2013-05-30 10:59	2013-05-30 11:59	ii Go)
- Enorman	and a second second	ومستوسية والمستوجي		£

Set a Relative Time Period

Use the **Date** list to select a relative time period; options range from "Last 5 Minutes" to "Last 2 Weeks." If you select a time period of less than 30 minutes, then the UI will auto-refresh and retrieve the latest historical data at the end of the collector aggregation interval.

Set an Absolute Time Range

Use the calendar controls to select an absolute time range. Set the dates (in the form "YYYY-MM-DD HH:MI:SS") and click **Go** to display data for this range.

View Historical Performance Data on the Activity window

The AppDynamics for Databases Activity window is the default window for the Database Platform. Most monitoring activities are reported here. The Activity window displays historical performance data that is retrieved from the repository. The In database Over Time panel shows the following:

- Executions: the number of SQL statements that were executed during that time period.
- In *database*: the total time spent executing those statements during that time period.



Set the View on the Activity Window

Click **View** on the Activity tab to change what appears in the Displayed rows section. The SQL View is the default.

The views available are database dependent. On each view of the Activity tab, you will see the In *database* Over Time and Top Wait States charts along with the Total Time in *database* and Total Executions metrics. When coming to a view by drilling down from another view, the entities are filtered to show only those related to the selection on the previous view.

For purposes of illustration, the Oracle database is used in this section.

5	Date: 🔻	2014-05-13 11:22 🛅 2014-05
	View: ၂၈	SQL
ł	Blocked S	Sessions
	Clients	
ž	Data Files	s
-	Modules	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1
244	Programs	5
Ì	Schemas	1
ł	Sessions	
1	SQL	
1	SQL Grou	iped
ł	Users	

Blocked Sessions View

The Blocked Sessions view shows sessions that hold an exclusive lock on an object that is not released before another session tries to update the same objects. The second session is blocked until the first one completes its updated. Blocked sessions can make the application look like its hung. It's important to identify blocked sessions to improve your application performance and avoid as many blocking sessions as possible.



Drill down from an entity in the Blocked Sessions view, to the Sessions View where you can see details pertaining only to the selected blocking session. The displayed rows will be filtered to display only the sessions related to that blocked session.

Clients View



In the Clients view, you can see details of all clients that ran in the selected time period.

Drilling down into an clients entity on the Clients view displays the SQL View where you can see the SQL running by the selected client during that time period. The displayed rows will be filtered to display only the SQL related to that client.

Data Files View

In the Data Files view, you can see details of all the database data files that were accessed in the selected time period.



Modules View

In the Modules view, you can see details of all modules that ran in the selected time period.



Drilling down into an modules entity on the Modules view displays the SQL View where you can see the SQL running by the selected module during that time period. The displayed rows will be filtered to display only the SQL related to that module.

Programs View



In the Programs view, you can see details of all programs that ran in the selected time period.

Drilling down into an programs entity on the Programs view displays the SQL View where you can see the SQL running by the selected program during that time period. The displayed rows will be filtered to display only the SQL related to that program.

Schemas/Databases View

Select **Schema** from the View list to view the Databases view, where you can see details of all databases that were in use in the selected time period.



Drilling down into a database entity on the Database view displays the SQL View where you can see the SQL running by the selected database during that time period. The displayed rows will be filtered to display only the SQL related to that database.

Sessions View

In the Sessions view, you can see details of all sessions that ran in the selected time period.



Drilling down into a session entity on the Session view displays the SQL Grouped View where you can see the groups of related SQL that ran by the selected session during that time period. The displayed rows will be filtered to display only the SQL groups related to that session.

SQL View



In the SQL view, you can see details of all Sessions that ran in the selected time period.

Clicking a SQL entity displays the SQL window that displays only details related to the specific SQL statement selected on the SQL view.



SQL Grouped View

The SQL Grouped feature within AppDynamics for Databases allows the grouping of similar queries by stripping their literals. In the case where a database based application does not use parameterized queries which utilize bind variables, then many similar queries will be executed which are effectively the same, only differing by their literals.

For example,

select * from table1 where col1 = `apple' and col2 = 1
is the same query as
select * from table1 where col1 = `pear' and col2 = 2

The AppDynamics SQL Grouped view would group these queries together and display them as:

select * from table1 where col1 = :s and col2 = :n

That is, it would substitute the string literal as :s and the numeric literal as :n

When coming to the SQL Grouped view from another view, such as the Sessions View, the SQL groups shown will be filtered to show only those SQL groups related to the session selected on the previous view. When unfiltered, all the SQL statements run during this time period display, the same as the SQL View.



Drilling down into a SQL statement on the SQL Grouped view displays the SQL View where you can see additional details of the selected SQL statement. The displayed rows will be filtered to display only details for the previously selected SQL statement.

Users View

In the Users view, you can see details of all users that were using databases on the selected host during the selected time period.



Drilling down into a user entity on the Users view displays the SQL View where you can see the SQL statements the user rant. The displayed rows will be filtered to display only details for the previously selected user.

Example Workflow Drilling Down Through Views



001 ID: haster2h1					
SQL ID: DCZIFZ3H3	sziwg				
SELECT count (1) count				
FROM Item Iti,	item 102				
a second second	and a substantion of the	and the second secon	a she ha al a she an	Sector 1	~~~~~ U .
L	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	\sim	man man	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
🛃 Explain Plan (E	rom cache)				8
- 101 SELECT STATE	EMENT (Optimizer Mode: ALL ROWS)				had at
= [1] SORT (AGO	GREGATE) { Rows: 1 Cost: N/A Bytes: N/A }				
		035 Buter: N/A 3			
[2] MERGE	JUIN (CARTESIAN) { ROWS: 99960001 COST: 49	ass bytes. In A j			
[2] MERGE [3] INDE	30IN (CARTESIAN) { ROWS: 99980001 COSt: 49 X (FAST FULL SCAN) SYS_C007179 { Rows: 9	9999 Cost: 7 Bytes: N/A }			
(2) MERGE (3) INDE (4) BUFF	UN (CARTESIAN) { Rows: 99960001 Cost: 49 X (FAST FULL SCAN) SYS_C007179 { Rows: 5 FER (SORT) { Rows: 9999 Cost: 49928 Bytes: 1	9999 Cost: 7 Bytes: N/A } V/A }			
(2) MERGE	JUIN (CARTESIAN) { Rows: 99900001 Cos:: 49 XX (FAST FULL SCAN) SYS_C007179 { Rows: 9 FER (SORT) { Rows: 9999 Cost: 49928 Bytes: 1 NDEX (FAST FULL SCAN) SYS_C007179 { Row	9999 Cost: 7 Bytes: N/A } V/A } vs: 9999 Cost: 5 Bytes: N/A }			
(2) MERGE (3) INDE (4) BUFF (5) IN	JUN (CARTESIAN) (Rows: 99900001 Cost: 49 X: (FAST FULL SCAN) SYS_C007179 (Rows: FER (SORT) (Rows: 9999 Cost: 49928 Bytes: N UDEX (FAST FULL SCAN) SYS_C007179 (Row S	9999 Cost: 7 Bytes: N/A } V/A } V/S } v/S \$9999 Cost: 5 Bytes: N/A }			
[2] MERGE [3] NDE [4] BUFI [5] M Number of steps: 6 Explained for Oracle-	JUNI (CARTESIAN) (ROWS: 99990001 COST: 49 X (FAST FULL SCAN) SYS_C007179 (Rows: 5 FER (SORT) (Rows: 9999 Cost: 49928 Bytes: 1 UDEX (FAST FULL SCAN) SYS_C007179 (Row 3 -demo-db in schema APPDY	9999 Cost: 5 Bytes: N/A } V/A } s: 5999 Cost: 5 Bytes: N/A }			
(2) MERGE (3) NDE (4) BUFI (5) M Number of steps: 6 Explained for Oracle-	JUNI (LARLESHAN) (ROWS 9990000 LOst as XY (FAST FULL SCAN) SYS_COUTYPI (Rows: FER (SORT) (Rows: 9999 Cost 49928 Bytes: N UDEX (FAST FULL SCAN) SYS_COUTYPI (Row 3 -demo-db in schema APPDY	9999 Cost: 5 Bytes: N/A } V/A } s: 9999 Cost: 5 Bytes: N/A }			
E2 MERGE	JUNI (CARLESHAN) (ROWS 9990000 Cost 49 V (FAST FULL SCAN) SYS_COUTYPI (Rows: FER (SORT) (Rows: 9999 Cost 49928 Bytes: h UDEX (FAST FULL SCAN) SYS_COUTYPI (Row 3 demo-db in schema APPDY cts	000 Cyta, INA } V/A } V/A } IS: 9999 Cost: 5 Bytes: N/A }			
E2 MERGE (3) NDE (5) IN (5) IN (5) IN Number of steps: 6 Explained for Oracle- Referenced Obje Tree Node	Uni (LAHRESHIN) (Nove: Septon) Loss: eq (X GAT FULL SCAI) SYS_COM719 (Rows: FER (SORT) (Rows 9999 Coat: 49928 Bytes: 1 (CAST FULL SCAI) SYS_COM719 (Rows) demo-db in schema APPDY cts Operation	000 0744 NA } WA	Object Name		

Indepth Analysis of SQL Statements

Once poorly-performing SQL has been identified by AppDynamics, you can click the statement for further analysis in the SQL View which displays comprehensive details of the statement.

To view statistics that are specific to a SQL statement, at the bottom of the **Activity** page, doubleclick the SQL statement. The **SQL Statistics** window appears.

Note: The Wait Time chart is available only for the Oracle database.



<pre>INSERT INTO TBL_BETS_UNFILLED (BET_TRANS_ID , BET_ID , ACCOUNT_ID , EVENT_ID , CURRENCY_ID , SELECTION_ID , STATUS_ID , BET_PRICE , BET_SIZE , BID_TYP , PROCESSED , PLACED_DATE , SUP_EVE_ID , SUBMISSION_ID , TO BE_MATCHED_TIMESTAMP , EXPO BET_SIZE , EXPO BET_PRICE , EXPO BSP_LIABILITY , BSP_BET_TYPE , BSP_LIABILITY , PERSISTENCE_TYPE) VALUES (SEQ_BET_TRANS_ID.NEXTVAL , :B20 , :B19 , :B18 , :B17 , :B16 , :B15 , :B5 , :B7 , :B14 , :B13 , :B12 , :B11 , :B10 , :B9 , CASE WHEN :B3 IS NOT NULL THEN 0 ELSE NVL(:B8 , :B7) END , NVL(:B6 , :B5) , NVL(:B4 , :B2) , :B3 , :B2 , :B1)</pre>	buffer busy waits enq: TX - index contention latch: enqueue hash chains Using CPU latch: cache buffers chains buffer deadlock latch: redo allocation	Top Wait State
an a fan fan fan an an fan fan fan fan f	eng: HW - contention	

Wait State Analysis for Oracle Only

For Oracle only. Wait state analysis is improved by the addition of a Wait Time chart as displayed, which allows you to see at a glance what functions are causing wait states in your environment and whether or not the time they consume is problematic.

View Historical %CPU Consumption

If, when you configured the AppDynamics for Databases collector, you enabled the Host Collector, then historical %CPU consumption metrics are available for the server hosting the monitored database.

ViewUp-to-the-Second Performance (Current window)

The AppDynamics for Databases Current window is the workspace where you can see an up-tothe-second view of performance. Rather than reading from the historical repository, the Current window connects directly to the monitored instance and requests current information using SQL queries.

The appearance of the Current window varies somewhat depending on the database platform type; however, it usually displays a graphical summary of current key performance indicators and a list of all currently connected database sessions or processes.



Database Statistics

The Statistics window displays historical information about key performance indicators of the database instance. The Statistics reported here vary by platform. For example, statistics reported for the IBM DB2 database include, Connections, Activity Type, Memory Key Performance Indicators, and Sort Performance.





To display historical graphs on any other collected statistic, use **Reports -> Database Statistics Report**.

ORACLE Sta	tistics Report				Other Reports:
Start Time	End Time	Aggregate	Database	Chart Multi-Series Line Chart 💠	Statistics bytes received via SQL*Net from client
Run Report					bytes received via SQL*Net from dblink bytes sent via SQL*Net to client bytes sent via SQL*Net to dblink CPU used by this session execute count index fast full scans (direct read) index fast full scans (full)
elect one or more st	atistics from the Statistics li	stbox.			

You can configure AppDynamics for Databases to collect tens or even hundreds of individual statistics. Add statistics using the **Setup -> Stats** window.

Main M	enu Setup	License	Archive	Groups	SQL	Stats	Waits	Mappings	Saved Data		ب
APP	NAMIC	CS	Setup Databas	se Statistics I	Manageme	ent					
Manag	e Statistics										
Add St	atistic										
Displayed	rows: 734										Export
[Show 💌											[Show 💌
Platform	Statistic	Name				Descriptio	on			Туре	Collect? D
IBMD62	ACTIVE_SORTS		The numbe	The number of sorts in the database that currently have a sort heap allocated.							
IBMD62	AGENTS_TOP		At the data	base level, it is ti	he maximum r	number of agent	ts for all applic	ations.	[ACT 🔻	

Add New Sta	atistic
Database Type:	MySQL 🔻
Statistic Name:	N.D. This must avoid the match the same of an avoid bla definitio form the manifested statiform. Can the User Ovide for more details
Description:	N.B. This must exactly match the name of an available statistic from the monitored platform. See the user Guide for more details.
Statistic Type:	Actual
	N.B. If the counter value is the actual value you would like to collect then select "Actual".
	If the counter value is continuously incrementing, then select "Cumulative", and the Collector will monitor the difference in value between samples.
	Add Statistic

The following table lists the source of statistical data. Any new statistic name must match exactly to the returned value from the source.

Database	
DB2	additional statistics not currently supported for this platform
mongoDB	additional statistics not currently supported for this platform
Microsoft SQL Server 2000	mastersysperfinfo
Microsoft SQL Server 2005 and above	sys.dm_os_performance_counters
Microsoft SQL Azure Database	additional statistics not currently supported for this platform
MySQL	show status

Oracle	v\$sysstat
PostgreSQL	additional statistics not currently supported for this platform
Sybase ASE	master.dbo.sysmonitors

Database Objects

The Objects window contains links to meta-data about the database configuration and schema by connecting directly to the monitored instance and requesting current information using SQL queries.

	ORACLE	Current Activity Statistics Objects	Host:
	Database Objects Users	orders	
	Parameters	Host	ip-10-99-37-110
O	Change	SID	XE
	Current Statistics	Version	11.2.0.2.0
	System Waits	Startup Time	2013-01-18 18:41:54.0
		Status	OPEN
1		Instance Role	PRIMARY_INSTANCE
{			
1			

The Objects window also contains information about configuration changes within the database instance since the collector began monitoring.