Examining Precise for SQL Server performance over time

This section includes the following topics:

- About the Activity tab
- How the Activity tab is structured
- About the entities you can examine
- How the Activity tab can help you identify performance problems

About the Activity tab

The Activity tab allows you to analyze performance behavior over time or within a given time period. When investigating a performance issue, the Activity tab is an excellent place to start. You can easily identify the performance profile of your SQL Server instance and determine who is executing which application, what they are waiting for and how many resources are they consuming. This is the critical information you need to identify the top consumers of your SQL Server Instance and its major latencies. By focusing on the top consumers and latencies (wait states) you can maximize the tuning impact on your system while minimizing the time it takes to obtain results.

The Activity tab is also very useful for a scale-up exercise. Too frequently companies add CPUs to an existing server in their attempt to improve performance. Although occasionally successful, step one of any scale up activity should be to identify the major latencies in the SQL Server Instance and resolve them. Only once the major wait states are resolved, will a scale-up actually be capable of improving performance. Otherwise, you may be simply wasting money and resources.

The information displayed in the Activity tab is derived from historical data and enables you to access information for different time intervals at varying levels of detail. You can analyze what happened at a specific time, pinpoint problematic time periods, and drill down to analyze a bottleneck that caused a performance problem.

In addition, if you are running SQL Server 2005, you can now view information collected by the SQL Server 2005 procedure cache. This data contains information on each plan cached and provides information on their statements and statistics on the execution of the plan's statements. This information allows you to pursue a wider range of new performance tuning scenarios.

The Activity tab was designed to answer questions such as, "What happened yesterday, as well as a month ago and a year ago?" and "What has been the performance level of my applications over time."

Data in the Activity tab is available up to the last time slice that ended (information on the last minute is available in the Current tab). The Activity tab is usually the place to identify and investigate tuning problems.

When working in the Activity tab, you can choose to work on a specific instance or to view information on all your instances, in the application.

> You usually switch to the Activity tab, after focusing on an instance in the Dashboard tab, or by launching Precise for SQL Server from another product (Precise for SAP, Precise for Microsoft .NET, Insight, Alerts, or Report Manager). In these cases, the product automatically focuses on the relevant instance.

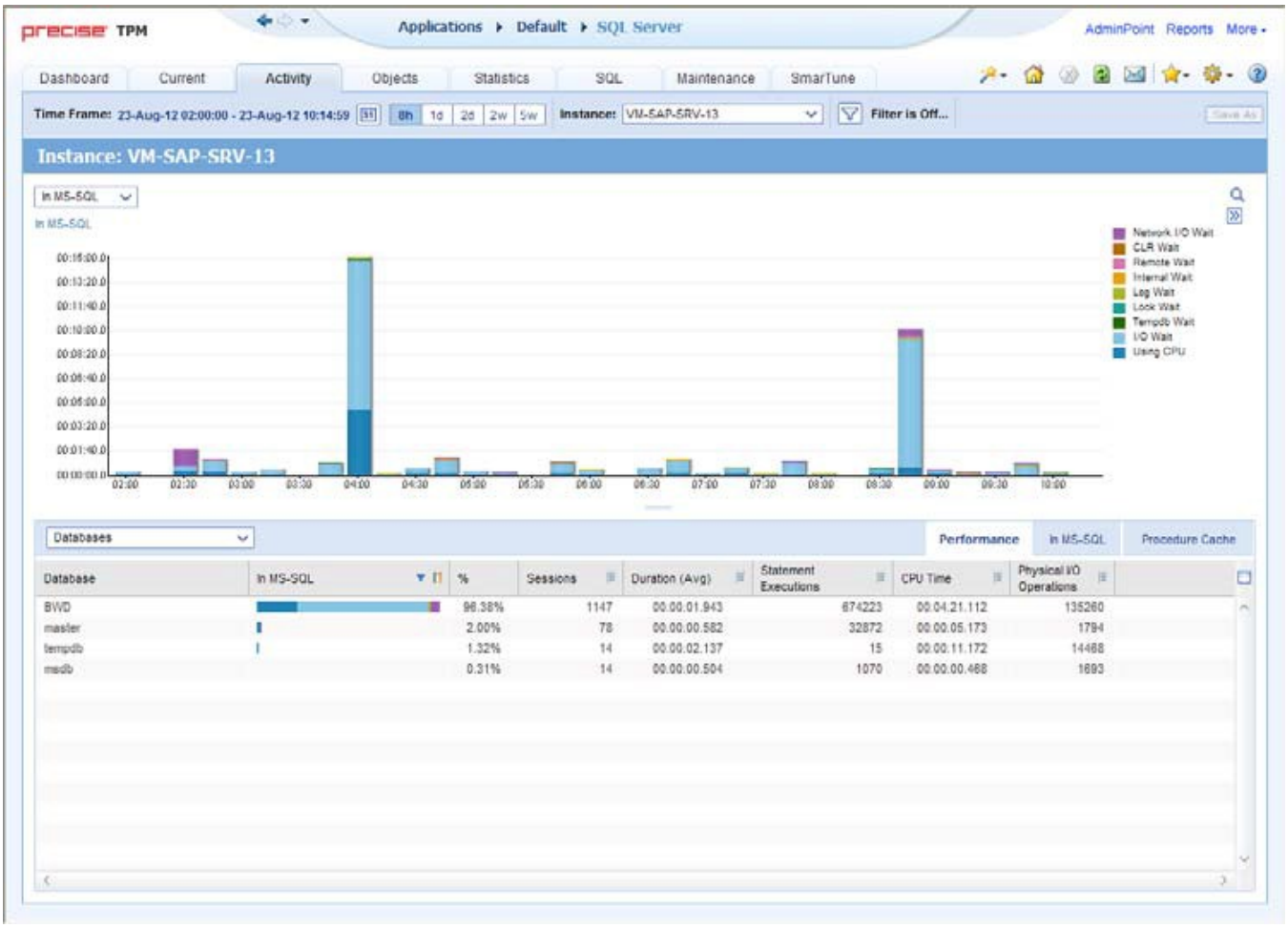
When you don't select a specific instance to view in-context, a summary of the performance information of All Instances in the application is displayed. You can analyze the average behavior of the entire SQL Server Tier. You can then identify the instance which consumes the most resources in your system and focus on it to further your investigation.

When you focus on a specific instance (either as a result of launching or from the SQL Server Tier view) you can also identify which is the dominant resource consumed, in terms of CPU, I/O lock, etc. By identifying your system's wait type you can formulate an idea as to what kind of solutions will help you improve your system.

> If you are working with SAP, PeopleSoft, or Siebel, and ERP support is installed, the Activity tab displays information on the historical SQL Server activity in relation to the ERP application and users, and specific information on executed views and transactions.

How the Activity tab is structured

Figure 1 Precise for SQL Server Activity tab structure



The Activity tab displays information on various entities and their associated entities. The information displayed when you launch the Activity tab depends upon where you are launching from.

For example, when you launch the Activity tab from the Dashboard tab, one of the following occurs, depending upon what you selected in the All SQL Server Instances table:

- If you selected an instance row, the focus in the Activity tab is on the selected instance.
- If you selected the All row, the focus in the Activity tab is on the Tier level.
- If you launch to the Activity tab using the launch icon in the Findings table in the SQL, Dashboard, or SmarTune tabs, you launch with the selected entity, in context.

When you access the Activity tab from other tabs, the entity, time frame, views, and associations that were in place the last time you visited the tab are displayed (similar to clicking the Back button in the Precise toolbar).

When you launch to another tab, from within the Activity tab, the entity and time frame you currently selected are taken into account. The tab you launched to will automatically display information in reference to the entity and time frame (if the entity exists in the tab you launched to). This is called in-context navigation and enables you to focus on other aspects of the selected entity to further your analysis.

The information on the selected entity and its associated entities are displayed in two areas of the tab—the Main area and the Association area.

When you perform an action on an element in the lower area (in this case the Association area) the information displayed in the upper area (in this case the Main area) will change to reflect the action you requested. Accordingly, you can control the tab by:

- Keeping the entity in the Main area but selecting a different set of entities to view in the Association are, by selecting an entity from the Association controls list.
- Switching to a different entity by drilling to another entity from the Association area.
- Switching to a different view in the Main area, thereby displaying additional information the selected entity.

About the Main area in the Activity tab

The Main area shows comprehensive information on the selected entity. You can choose from several views to examine the entity from different angles. For example, you can focus on an In MS-SQL overtime graph for an Instance entity to observe its resource consumption over time or you can focus on the Locking view to investigate all the locks performed on the instance.

See About drilling down in context, About the Precise bar, About the Main area, About the Association area, and Focusing on information in overtime graphs.

About navigating in the Activity tab

The name of the entity you selected is displayed in the tab heading, which serves as a point of orientation. The highest level entity that you can view information for in the Activity tab is the Tier.

Moving the pointer over the Tab heading displays a ToolTip with navigational information that traces the path you have already taken in your investigation.

About the Association area in the Activity tab

The Association area displays relevant information on the entities associated with the selected entity (displayed in the Tab heading in the Main area) in a table format. For example, it is possible to associate to Databases, Statements and Logins that are related to a specific Instance, by selecting an entity from the Association controls list. Notice that some entities, show additional information on the associated entities in different tabs. The tabs are located above the Association area table. Clicking on a tab displays different table columns showing different information for the associated entities. For example, the following tabs show additional information for the Statement entity:

- Performance
- In MS-SQL
- Statistics
- Procedure Cache (SQL Server 2005)
- Plan

If you want to view additional information on an associate entity, drill down to it, by selecting the entity's row. A drilldown affects the entire tab. When you drill down to another entity, the Tab heading reflects the new selection, the Main area displays information on the selected entity, and the Association area displays the entities that are now associated with the newly selected entity.

For example, when you want to drill down to an associated entity, such as a specific program, to view additional information, choose Programs from the Association controls. Note that the Tab heading and the Main area remain unchanged. Click the row of the program you want to view detailed information for. The Tab heading indicates the newly selected program; the Main area displays In MS-SQL information on the program you drilled down to, and the Association area shows information on statements executed by this program.

See About drilling down in context, About the Precise bar, About the Main area, About the Association area, and Focusing on information in overtime graphs.

About the entities you can examine

The Activity tab displays information on different entities. This section provides an overview of all entities, their meaning, and their views. By default, all entities open in the In MS-SQL view, with the exception of the Statement, Batch, Collapsed Statement and Collapsed Batch entities which open by default in the Overview view.

The following entities can be examined in the Activity tab:

- All Instances
- Instance and Collector Entities
- Statement, Batch, Collapsed Statement or Collapsed Batch Entities
- Session and Blocker Session Entities
- Locked Object Entity
- Table and Index Entities
- I/O Entities

Some of these entities have been grouped together logically because they can be described similarly. They appear separately when viewed in the GUI.

About the All Instances entity

A Tier entity displays the resource consumption of all the instances in the application in the following views:

- Overview
- In MS-SQL
- Internal wait
- Instance grouping

About viewing a summary of system performance and resource consumption for All Instances

The Overview displays a summary of the performance and resource consumption of all instances in the application, as follows:

- **Instances**. Displays the number of Instances defined in the Tier.
- **In MS-SQL breakdown table**. Displays the resource consumption breakdown of all the instances over the selected time period, as follows:

> The table only displays the states the selected entity was in. For example, if the selected entity was not waiting for an I/O wait during the selected time period, the table will not display I/O wait parameters.

About viewing In MS-SQL data for All Instances

This view displays an overtime graph of the total time spent In MS-SQL for all instances in the SQL Server Tier.

About viewing the Internal Wait time experienced by All Instances

The Internal Waits view shows a breakdown of the internal wait counter in an overtime graph. The internal wait counters are broken down into the following subsets:

- Buffer Pool
- Latch
- Parallel
- Distributed Transaction Coordinator (DTC)
- DB Mirror
- Profiler
- Memory
- Backup
- Other Internal waits

See About internal waits.

About viewing Instance Grouping for All Instances

Displays instance grouping information for the selected SQL Server Tier. It is possible to categorize instances into groups and observe their tuning counters, such as their In MS-SQL breakdown, according to the groups you defined. This option is useful if there are many instances in the same application.

For an explanation on how to define instance groups, see the Precise Administration Guide.

About Instance and Collector entities

The Instance entity and the common Collector entities display the resource consumption of the entire instance (or the entity selected). When focusing on an instance, or collector entity, such as a program, you can examine how it is performing within the selected time period.

> The information displayed for certain entities may differ slightly than that described, according to what is relevant to the specific entity. For example, if you have navigated to a storage entity (such as a database file), only the I/O wait information for the selected entity is displayed. If you have navigated to a locked object, only the Lock wait information is displayed.

The following table describes which Collector entities can be examined in the Activity tab.

Table 1 Collector entities that can be examined in the Activity tab

Collector Entity	Description
Database	Displays information on an activity in reference to a specific database.
Program	Displays information on an activity generated from a specific program. Displays 'N/A' if the program_name column in sysprocesses is null. If ERP is installed, the program is updated with the ERP data. For more information, see extended ERP data.
Login	Displays information on an activity generated by a specific login. Displays 'N/A' if the login name column in sysprocesses is null. If ERP is installed, the login is updated with the ERP data. For more information, see extended ERP data.
User	Displays information on an activity generated by a specific user in the database in use.
Machine	Displays information on an activity generated from a specific workstation. Displays 'N/A' if the hostname column in sysprocesses is null. If InterPoint for ERP is installed, the machine is updated with the ERP data. For more information, see extended ERP data.

Work Type	The Precise for SQL Server Collector divides SQL Server activities into several groups. The Work Type entity displays performance information on each group. The following groups are available: • **User Application**. Activities generated by application components that are not one of the following: SQL agent, COM+, PeopleSoft, Siebel, SAP. • **SQL Agent Job**. Activities generated by the SQL Agent jobs. • **COM+**. Activities generated by a COM+ application (displayed only if Interpoint for COM+ is installed). The following groups are only displayed if InterPoint for ERP is installed: In SAP: ○ **Dialog**. Activities generated from SAP Dialog work process. ○ **Update**. Activities generated from SAP Update work process. ○ **Spool**. Activities generated from SAP Spool work process. ○ **Background**. Activities generated from SAP Background work process. ○ **Enqueue**. Activities generated from SAP Enqueue work process. In PeopleSoft: ○ **Interactive**. Activities generated from the PeopleSoft Interactive panel. ○ **Batch**. Activities generated from the PeopleSoft batch. In Siebel: ○ **Interactive**. Activities generated from the Siebel interactive application. ○ **Batch**. Activities generated from the Siebel Batch process. ○ **Background**. Activities generated from the Siebel Background process.
COM+	For more information, see extended ERP/COM+ documentation.
ERP Data	For more information, see extended ERP/COM+ documentation.
Job	Displays information on an activity, with reference to a specific job.

See Precise for SQL Server agents.

The following views are available for the Instance and Collector entities:

- Overview
- In MS-SQL
- Locking
- Internal Waits
- Procedure Cache
- Overall Activity
- Scalability
- System Scalability
- Availability

About getting an overview of Instance and Collector data

The Overview displays information on the performance and resource consumption of the collector entities. The following information is displayed:

- **Load**. Displays information on the following session-related items, as follows:
  - **Sessions**. Number of sessions that occurred during the specified time period of the selected collector entity. For example, if you are viewing the Login entity, this value represents the number of sessions that are connected to the selected Login.
  - **Duration (Avg)**. Average duration for a single session, in the selected collector entity, during the selected time period, calculated by dividing the Duration by the number of Sessions.
  - **Duration (Summed)**. Total amount of time that elapsed for all of the selected collector entity's sessions, during the selected time period.
  - **Statement Executions**. Number of statements executed within the selected time period that were generated by the selected collector entity's sessions.
  - **Statement Duration (Avg)**. Average duration for a single statement execution, in the selected collector entity, during the selected time period. This value is calculated by dividing the Duration (Summed) parameter by the number of statement executions.
- **Statistics**. Displays the following session-related statistics, as follows:
  - **CPU Time**. Cumulative CPU time for the collector entity or instance, during the selected time period. This counter is not accurate when SQL Server is configured to use fiber mode (lightweight pooling option).
  - **Physical I/O Operations**. Total number of physical I/O operations performed by all sessions connected to the database during the selected time period.
- **In MS-SQL breakdown**. Displays collector entity resource consumption (table format) and shows instance breakdown, within the selected time period, as follows:
  - **State**. Displays the In MS-SQL state.
  - **In MS-SQL**. Displays a graphical representation of the Time column.
  - **Time**. Amount of time the selected collector entity was in this state.
  - **%**. Percentage of time the selected collector entity was in this state.
    See About session states.

About viewing Instance and Collector In MS-SQL data

The In MS-SQL view displays the resource consumption of the selected collector entity over the selected time period. This graph enables you to analyze performance trends for the collector entity, over time.

See About session states and Focusing on information in overtime graphs.

About viewing Instance and Collector locking data

The Locking view displays the blocking tree and the lock wait duration broken down into lock levels (such as, Table, Page, and Key), for the selected entity, over the selected time period. The following information is displayed:

- **Locks**. Displays which sessions were involved in a blocking situation (either locking or blocking). It is important to understand which sessions are involved in each blocking chain, as there may be several sessions involved at the same time. It is then important to determine which session is blocking others, which object is being blocked, and what type of lock has been applied.
  A yellow lock indicates that the session is currently blocking another session but is not itself blocked.
  A red lock indicates the session is currently blocked by another session. Note that a session may also block other sessions while being blocked by a different session.

- **Lock wait graph**. The Lock Wait graph shows a breakdown of the Lock Waits, and when they occurred, over the selected time period.

See Focusing on information in overtime graphs.

About the lock tree structure

To gain a better understanding of the lock tree let's take a look at its structure. Notice that the session located in the root of each chain is the session that blocks all others. Each session directly blocks all sessions positioned one level beneath it.

The following information is displayed in the Lock tree:

- start time of the lock
- duration of the lock
- the SPID of the sessions involved in the lock
- the program name and the client process ID of the session.

Focusing on a blocked session branch in the lock tree will expose the SQL text performed by the session. The lock type is displayed in the Lock Breakdown table, which is displayed to the right of the Lock Trees.

Every time a change occurs, a new lock tree is opened, as for example, when a new session enters into the lock. It is therefore possible that there will be sub-trees of several locks that may appear identical.

About viewing internal waits information for Instance and Collector entities

The Internal Waits view displays the following internal waits, indicating that the session is waiting for an internal resource to be freed:

- **Other internal wait**. Aggregates the following types of waits:
  - Full text waits
  - HTTP waits
  - Query notifications
- **Backup wait**. Includes the wait type that commonly occurs when a Backup command is performed.
- **Memory wait**. Includes ten types of waits that indicate that a session is waiting for memory to be allocated to it.
- **Profiler wait**. Aggregates a number of states associated with the SQL Profiler and lets you see how much of the database resources it consumes.
- **DB mirror wait**. Aggregates the new waits that occur when DB mirroring is performed, such as the waits that occur if the communication layer used by DB mirroring becomes backlogged.
- **DTC wait**. Aggregates waits that occur when Distributed Transaction Coordinator (DTC) sessions have to wait for each other.
- **Parallel wait**. The session is waiting for one of its sub-threads to complete its operation.
- **Latch wait**. The session is waiting for an internal lock to be released.
- **Buffer pool wait**. Groups together the events that show contention on pages in the buffer pool.

About viewing procedure cache information for Instance and Collector entities

The Procedure Cache is the part of the SQL Server memory pool that stores execution plans for batches, stored procedures and triggers. The Procedure Cache view provides the following information on an execution plan:

- Load
  - Elapsed time
  - Execution count
  - Avg duration
- Statistics
  - Recompilations
  - CPU time
  - CLR time
- I/O
  - Logical reads
  - Logical writes
  - Physical reads
- Avg Duration vs. Execution Count graph

    - Graph showing the amount of time it took to complete the execution of a plan versus the number of times the plan was executed.
- I/O graph
    - Graph showing the date and time that physical reads, and logical read/writes occurred.

About viewing overall activity of Instance and Collector entities

The Overall Activity view displays the duration of the activity performed by the selected collector entity over the selected time period, broken down into the following components:

- **Waitfor Command**. The amount of time applications were waiting because they were waiting for command.
- **Parallel Exchange Wait**. The amount of time the thread in a parallel session is waiting for data exchange from another thread.
- **Request Wait**. The amount of time applications were in a request wait state, defined as the state wherein the server is waiting for a client request.
- **In MS-SQL**. Represents total amount of time spent in SQL Server. See Focusing on information in overtime graphs.

About viewing Instance and Collector Scalability

The Scalability view displays two overtime graphs that allow you to examine the scalability of the selected collector entity, as follows:

- **Sessions duration (Avg) vs. sessions**. Compares the average duration of the selected collector entity with the number of sessions, over the selected time period. For example, if the selected collector entity is Login the graph will show the number of sessions connected with the selected login versus the average duration of each session over time. Use this graph to analyze if the average session duration increases when the number of sessions increase.
- **I/O wait vs. physical I/O Operations**. Compares the amount of time the collector entity waited for the I/O to be completed with the actual physical I/O requests performed on behalf of the collector entity. Use this graph to analyze how an increase of physical I/O requests affects the I /O wait of the collector entity.

See Focusing on information in overtime graphs.

About viewing Instance and Collector System Scalability

The System Scalability view displays two overtime graphs that allow you to examine the scalability of the selected collector entity when analyzing OS metrics, as follows:

- **In MS-SQL vs. SQL server CPU usage (%)**. Compares the In MS-SQL of the selected collector entity with SQL Server CPU usage over the selected time period. For example, if the selected collector entity is Login the graph will show the In MS-SQL time consumed by the selected login and SQL Server CPU utilization over time. Use this graph to evaluate the correlation between collector entity performance and the SQL Server CPU usage. For example, you can determine if high CPU utilization is a result of collector entity's activity.
- **Physical I/O Operations vs. queue length (Avg)**. Compares the physical I/O requests performed on behalf of the collector entity with the average queue length of the available disks over the selected time period. Since an increase in queue length affects application performance, this graph enables you to analyze how an increase in physical I/O requests affects queue length.

See Focusing on information in overtime graphs.

About viewing Instance and Collector Availability

The Availability view displays the percentage of time the SQL Server instance was available, that is, it met the SLA Availability requirements configured in the SLA settings. A green bar represents the amount of time the instance availability met with SLA requirements and a Red bar represents the amount of time the instance availability was lower than SLA requirements.

> During planned downtime periods the instance availability is displayed as if it was available. This view is available only for instance and database entities.

The following information is displayed in this view:

- **Availability**. Displays the percentage of time the SQL Server was available (including planned down time; for more information see the Precise Administration Guide).
- **Unavailability times**. Displays how long the SQL Server was down. If the Precise for SQL Server Collector was down, the status will be labeled "Unknown Time."
- **Log**. Provides general information regarding each period that the SQL Server was down.
  Provide information on the status of SQL Server instance, over the selected time period. If the instance was down or if the status was unknown (the Collector was down) during the selected time period, only information on the unavailability of the instance is displayed (time range of the unavailability period).

See Focusing on information in overtime graphs.

About the Instance and Collector entities Association area

Additional information on performance, In MS-SQL activity, and the procedure cache, for the instance and collector entities, is displayed in the Association area tabs.

Clicking a tab shows additional information on the entities displayed in the Association area table. The rows of the table remain the same; the columns change to display additional details on the entities listed in the table.

About viewing performance data for Instance and Collector entities

The following table describes the additional information displayed in the Performance tab in the Association area.

Table 2 Information displayed in the Performance tab

Column	Description
Entity name (such as instance, database, etc.)	Displays the name of the entity.
In MS-SQL	Displays entity resource consumption (default: stacked bar chart) and shows entity breakdown, over the selected time period.
%	Displays the percentage of the total In MS-SQL.
Sessions	Number of sessions that occurred during the selected time period of the selected entity.
Duration (Avg)	Average duration for a single session, in the selected entity, during the selected time period, calculated by dividing the Duration by the number of Sessions.
Statement Executions	Number of statements executed within the selected time period that were generated by the selected entity's sessions.
CPU Time	Cumulative CPU time for the entity during the selected time period. This value is calculated by the Collector as follows (this value is an approximation)—the CPU Time field in sysprocesses is sampled at the start point and end point. The value sampled at the start point is subtracted from the value sampled at the end point resulting in the CPU Time reported. The start and end points are determined according to values that have changed. For example, a session is sampled by the Collector for five seconds, but the start and end points were calculated according to when a value has changed. As a result, the calculated CPU Time in this example is only three out of the sampled five seconds since this is takes into consideration when the value has changed.

About viewing In MS-SQL data for Instance and Collector entities

The In MS-SQL tab contains the In MS-SQL counter and the breakdown of the In MS-SQL into its states. The In MS-SQL is displayed with a normalized view while all the other counters display the time each state was consumed. See About session states.

About viewing procedure cache data for Instance and Collector entities

The Procedure Cache is the part of the SQL Server memory pool that stores execution plans for batches, stored procedures and triggers. Detailed information on the execution plan run of a specific instance is displayed in the Procedure Cache tab.

The following table describes the information displayed for execution plans in the Procedure Cache tab.

Table 3 Information displayed in the Procedure Cache tab

Column	Description
Entity name (such as Instance, Database)	Displays the name of the entity.
Elapsed Time	Amount of time it took to complete the executions of the plan.
Execution Count	Number of times the plan was executed.
Avg. Duration	Average amount of time it took to execute the plan, calculated as total elapsed time divided by the execution count.
Recompilations	Number of times a plan was recompiled while it remained in the procedure cache.
CPU Time	Amount of CPU time consumed by the executions of the plan.
CLR Time	Time consumed inside CLR objects by the executions of the plan.
Logical I/O	Number of logical reads and writes performed by the executions of the plan.
Physical Reads	Number of physical reads performed by the executions of the plan.

About Statement, Batch, Collapsed Statement or Collapsed Batch entities

When an application component (such as Database or Login) consumes resources in SQL Server, it is actually executing a statement that is consuming resources. Tuning performance problems typically involves finding and tuning offending statements.

The following table describes which performance and resource consumption information is displayed for the statement, batch, collapsed statement and collapsed batch entities.

Table 4 Statement, Collapsed Statement, Batch and Collapsed Batch entities

Entity	Description
Statement	Displays information on the activity of a specific TSQL statement. A statement is always part of a batch. When an encrypted stored procedure is encountered the statement displays, "Encrypted Text."
Batch	Displays information on the activity of a specific batch. A batch can either be a stored procedure, stored function, trigger, or an ad hoc batch. If it is a stored object, the name of the stored objects are displayed.
Collapsed statement	Represents similar statements that differ in their parameters (literals). In a collapsed statement, the literals are replaced with parameter markers. The collapsed statement lets you analyze the overall resource consumption of similar statements.
Collapsed batch	Similar to collapsed statement a collapsed batch represents similar batches that differ in their parameters (literals). In a collapsed batch, the literals are replaced with parameter markers. The collapsed batch lets you analyze the overall resource consumption of similar batches.

The information displayed for certain entities may differ slightly than that described, as is relevant for the specific entity. For example, if you have navigated to a storage entity (such as a database file), only the I/O wait information for the selected entity is displayed. If you have navigated to a locked object, only the Lock wait information is displayed.

See About Statement identifiers.

The following views are available for the Statement, Batch, Collapsed Statement and Collapsed Batch entities:

- Overview
- Text
- In MS-SQL
- Locking
- Internal Waits
- Procedure Cache
- Scalability
- System Scalability

About getting an overview of Statement, Collapsed Statement, Batch and Collapsed Batch entities

The Overview displays information on the entity's activity, such as its number of executions and average statement duration. In addition the resource consumption breakdown and the text of the statement or batch are also displayed, as follows:

- **Load**. Displays information on the following session-related items:
  - **Database**. Identifies the statement's database.
  - **User**. Identifies the statement's parsing user.
  - **Statement Executions**. Number of times the Collector identifies that the statement was executed during the selected time period.
  - **Statement Duration (Avg)**. Displays the average duration of each statement execution within the selected time period, calculated by dividing the Duration (Summed) by the number of Statement Executions.
  - **Statement Duration (Summed)**. Total elapsed time required to run all executions of the selected statement or batch during the selected time period.

- **Statistics**. Displays the following statement-related statistics:
  - **CPU Time**. Cumulative CPU time for the entity during the selected time period. This value is calculated by the Collector as follows (this value is an approximation)—the CPU Time field in sysprocesses is sampled at the start point and end point. The value sampled at the start point is subtracted from the value sampled at the end point resulting in the CPU Time reported. The start and end points are determined according to values that have changed. For example, a statement is sampled by the Collector for five seconds, but the start and end points were calculated according to when a value has changed. As a result, the calculated CPU Time in this example is only three out of the sampled five seconds since this is takes into consideration when the value has changed.
  - **Physical I/O Operations**. Total number of physical I/O requests during statement execution, within the selected time period.
  - **Parallel Sessions (Min)**. Certain statements are executed using parallelisms. This counter displays the minimum number of threads used to execute the statement and is calculated by examining the `ecid` column in `sysprocesses`. Check this counter to verify that the SQL Server is implementing the best execution plan for the statement.
  - **Parallel Sessions (Max)**. Certain statements are executed using parallelisms. This counter displays the maximum number of threads used to execute the statement and is calculated by examining the `ecid` column in `sysprocesses`. Check this counter to verify that the SQL Server is implementing the best execution plan for the statement.
- **In MS-SQL breakdown**. Displays resource consumption broken down into states, such as, Using CPU, I/O Wait, Lock Wait, etc., within the specified time period. This allows you to identify the highest resource consumers.
- **Text**. Displays statement or batch text (formatted for easy readability).
  When viewing a batch entity, the entire text of the ad-hoc batch or stored object is displayed.

When viewing a statement, only the part of the batch that generates the statement is displayed.
When viewing a collapsed statement or batch, the literals are replaced with parameter markers.

About viewing Statement, Collapsed Statement, Batch and Collapsed Batch text

The Text view displays the SQL Text of the statement or batch.

About viewing In MS-SQL data for statements, collapsed statements, batches and collapsed batches

The In MS-SQL view displays the resource consumption of the selected entity over the selected time period. This graph enables you to analyze performance trends for the entity, over time.

See Focusing on information in overtime graphs.

About viewing Statement, Collapsed Statement, Batch and Collapsed Batch entities locking

The Locking view displays the blocking tree and Lock Wait duration of the selected entity over a specified time period broken down into lock levels such as, Table Lock, and Page Lock. The following information is displayed:

- **Locks**. Displays which sessions were involved in a blocking situation (either locking or blocking). It is important to understand which sessions are involved in each blocking chain, as there may be several sessions involved at the same time. It is then important to determine which session is blocking others, which object is being blocked, and what type of lock has been applied.
  A yellow lock indicates that the session is currently blocking other sessions but is not itself blocked.
  A red lock indicates the session is currently blocked by another session. Note that a session may also block other sessions while being blocked by a different session.
- **Lock wait graph**. The Lock Wait graph shows a breakdown of the Lock Waits, and when they occurred, over the selected time period.

See Focusing on information in overtime graphs.

About the lock tree structure

To gain a better understanding of the lock tree let's take a look at its structure. Notice that the session located in the root of each chain is the session that blocks all others. Each session directly blocks all sessions positioned one level beneath it.

The following information is displayed in the Lock tree:

- start time of the lock
- duration of the lock
- the SPID of the sessions involved in the lock
- the program name and the client process ID of the session.

Focusing on a blocked session branch in the lock tree will expose the SQL text performed by the session. The lock type is displayed in the Lock Breakdown table, which is displayed to the right of the Lock Trees.

Every time a change occurs, a new lock tree is opened, as for example, when a new session enters into the lock. It is therefore possible that there will be sub-trees of several locks that may appear identical.

About viewing internal waits data for statements, collapsed statements, batches and collapsed batches

The Internal Waits view displays the following internal waits, indicating that the session is waiting for an internal resource to be freed:

- **Other internal wait**. Aggregates the following types of waits:
  - Full text waits
  - HTTP waits
  - Query notifications
- **Backup wait**. Includes the wait type that commonly occurs when a Backup command is performed.
- **Memory wait**. Includes ten types of waits that indicate that a session is waiting for memory to be allocated to it.
- **Profiler wait**. Aggregates a number of states associated with the SQL Profiler and lets you see how much of the database resources it consumes.
- **DB mirror wait**. Aggregates the new waits that occur when DB mirroring is performed, such as the waits that occur if the communication layer used by DB mirroring becomes backlogged.
- **DTC wait**. Aggregates waits that occur when Distributed Transaction Coordinator sessions have to wait for each other.
- **Parallel wait**. The session is waiting for one of its sub-threads to complete its operation.
- **Latch wait**. The session is waiting for an internal lock to be released.
- **Buffer pool wait**. Groups together the events that show contention on pages in the buffer pool.

About viewing procedure cache data for statements, collapsed statements, batches and collapsed batches

The Procedure Cache is the part of the SQL Server memory pool that stores execution plans for batches, stored procedures and triggers. The Procedure Cache view provides the following information on an execution plan:

- **Load**. Elapsed time Execution count average duration.
- **Statistics**.
  - Recompilations
  - CPU time
  - CLR time

- **I/O**. Logical reads Logical writes Physical reads

- **Avg Duration vs. Execution Count graph**. Graph showing the amount of time it took to complete the execution of a plan versus the number of times the plan was executed.
- **I/O graph**. Graph showing the date and time that physical reads, and logical read/writes occurred.

About viewing scalability for statements, collapsed statements, batches and collapsed batches

The Scalability view displays two overtime graphs that allow you to examine the scalability of the selected collector entity over the selected time period. For example, you can determine whether the average statement duration of an entity's activity increased as a result of an increase in number of executions.

The following overtime graphs are displayed:

- **Statement duration (Avg) vs. executions**. Compares the average duration of the selected entity with the number of executions, over the selected time period. For example, if the selected entity is Statement, the graph will show the number of executions of the selected statement versus the average duration of each execution over time. Use this graph to analyze if the average statement duration increases when the number of executions increase.
- **I/O wait vs. physical I/O Operations**. Compares the amount of time the entity waited for the I/O to be completed with the actual physical I/O requests performed on behalf of the entity. Use this graph to analyze how an increase of physical I/O requests affects the I/O wait of the entity.

About viewing system scalability for statements, collapsed statements, batches and collapsed batches

The System view displays two overtime graphs that allow you to examine the scalability of the selected entity when analyzing OS metrics.

This information enables you to evaluate the correlation between the activities performed by the selected entity and the general load. For example, is the SQL Server CPU Usage mostly consumed by the selected statement?

The following overtime graphs are available:

- **In MS-SQL vs. SQL server CPU usage (%)**. Compares the In MS-SQL of the selected entity with SQL Server CPU usage.
  For example, if the selected entity is Statement the graph will show the In MS-SQL time consumed by the selected Statement and SQL Server CPU utilization over time. Use this graph to evaluate the correlation between entity performance and the SQL Server. CPU utilization represents an average of the utilization of all processes.

- **Physical I/O Operations vs. queue length (Avg)**. Compares the physical I/O requests performed on behalf of the entity with the average queue length of the available disks. Since an increase in queue length affects application performance, this graph enables you to analyze how an increase in physical I/O requests affects queue length.

See Focusing on information in overtime graphs.

About the Statement and Batch entities Association area

The following tabs are displayed in the Association area of the Statement and Batch entities:

- Performance
- In MS-SQL
- Statistics
- Procedure Cache
- Plan

Clicking a tab shows additional information on the entities displayed in the Association area table. The rows of the table remain the same; the columns change to display additional details on the entities listed in the table.

The Actual Plan column displays if the plan for statement or batch is an actual plan. This is available for SQL Server 2005 only.

About viewing performance data for Statement and Batch entities

The following table describes the additional information displayed in the Performance tab in the Association area.

Table 5 Information displayed in the Performance tab

Column	Description
	Click to launch the SQL tab with the selected entity, to continue your analysis of the selected statement or batch. When launching to the SQL tab from a batch entity, the launch is performed in the context of the batch that has the latest explain statement performed on it.
Entity Name	Displays the name of the entity (statement or batch).
In MS-SQL	Displays entity resource consumption (default: stacked bar chart) and shows entity breakdown, over the selected time period.
%	Displays the percentage of the total In MS-SQL.
Statement Executions	Number of statements executed within the selected time period by the selected entity.
Statement Duration (Avg)	Average duration for a single statement, in the selected entity, during the selected time period, calculated by dividing the Statement Duration by the number of executions.

Text	Displays the statement's text (formatted for easy readability).

About viewing In MS-SQL information for Statement and Batch entities

The In MS-SQL tab contains the In MS-SQL counter and the breakdown of the In MS-SQL into its states. The In MS-SQL is displayed with a normalized view while all the other counters display the time each state was consumed.

About viewing Statistics information for Statement and Batch entities

The following table describes the additional information displayed in the Statistics tab in the Association area.

Table 6 Information displayed in the Statistics tab

Column	Description
	Click to launch the SQL tab with the selected entity, to continue your analysis of the selected statement or batch.
Entity Name (such as Statement)	Displays the name of the entity.
CPU Time	Cumulative CPU time for the entity during the selected time period.
Physical I/O Operations	Total number of physical I/O requests performed by the selected entity during the selected time period.
Parallel Sessions (Min)	Minimum number of threads used to execute the statement or batch in parallel. This counter is calculated from the `ecid` column in `sysprocesses`. This enables you to verify that SQL Server is using the best execution plan for the current statement.
Parallel Sessions (Max)	Maximum number of threads used to execute the statement or batch in parallel. This counter is calculated from the `ecid` column in `sysprocesses`. This enables you to verify that SQL Server is using the best execution plan for the current statement.

About viewing Procedure Cache information for Statement and Batch entities

The Procedure Cache is the part of the SQL Server memory pool that stores execution plans for batches, stored procedures and triggers. Detailed information on the execution plan run of a specific statement, collapsed statement, batch or collapsed batch is displayed in the Procedure Cache tab.

The following table describes the information displayed for execution plans in the Procedure Cache tab

Table 7 Information displayed in the Procedure Cache tab

Column	Description
Entity name (such as Instance, Database)	Displays the name of the entity.
Elapsed Time	Amount of time it took to complete the executions of the plan.
Execution Count	Number of times the plan was executed.
Avg Duration	Average amount of time it took to execute the plan, calculated as total elapsed time divided by the execution count.
Recompilations	Number of times a plan was recompiled while it remained in the procedure cache.
CPU Time	Amount of CPU time consumed by the executions of the plan.
CLR Time	Time consumed inside CLR objects by the executions of the plan.
Logical I/O	Number of logical reads and writes performed by the executions of the plan.
Physical Reads	Number of physical reads performed by the executions of the plan.

About viewing Plan information for Statement and Batch entities

The following table describes the additional information displayed in the Plan tab in the Association area.

Table 8 Information displayed in the Plan tab

Column	Description
	Click to launch the SQL tab with the selected entity, to continue your analysis of the selected statement or batch. Explain data for the statement will be in extrapolated context to the batch.

Entity Name (Such as Statement)	Displays the name of the entity.
Database	Identifies the database on which the statement or batch is run.
User	Displays one of the SQL Server users by whom this statement was executed. If this statement is not part of a stored procedure, this is the user that will be used as the parsing user when explaining this batch.
Most Recent Plan	Indicates the last time the access path of the statement or batch was changed.
Last Show Plan	Indicates the last time the statement or batch was explained.
Cost	Indicates the total estimated cost of the statement's or batch's execution.
Table Scan	Indicates whether a table scan was performed in the execution plan (applicable to Statement entity only).
Clustered Index Scan	Indicates whether a clustered index scan was performed in the execution plan (applicable to Statement entity only).
clustered Index Seek	Indicates whether a clustered index seek was performed in the execution plan (applicable to Statement entity only).
Index Scan	Indicates whether an index scan was performed in the execution plan (applicable to Statement entity only).
Index Seek	Indicates whether an index seek was performed in the execution plan (applicable to Statement entity only).
Bookmark	Indicates whether a bookmark operation was performed in the execution plan (applicable to Statement entity only).
Sort	Indicates whether a sort operation was performed in the execution plan (applicable to Statement entity only).
Merge Join	Indicates whether a merge join operation was performed in the execution plan (applicable to Statement entity only).
Hash Join	Indicates whether a hash join operation was performed in the execution plan (applicable to Statement entity only).
Nestled Loops Join	Indicates whether a nested loops join operation was performed in the execution plan (applicable to Statement entity only).
Parallel Access	Indicates whether a parallel access operation was performed in the execution plan (applicable to Statement entity only).
Show Plan Error	Shows the error that occurred during the last explain process.

About the Collapsed statement and collapsed batch entities Association area

The following tabs are displayed in the Association area of the Collapsed Statement and Collapsed Batch entities:

- Performance
- In MS-SQL
- Statistics
- Procedure Cache

Clicking a tab shows additional information on the entities displayed in the Association area table. The rows of the table remain the same; the columns change to display additional details on the entities listed in the table. The information displayed in each tab is described below.

About viewing performance data for Collapsed Statement and Collapsed Batch entities

The following table describes the additional information displayed in the Performance tab in the Association area.

Table 9 Information displayed in the Performance tab

Column	Description
Entity Name	Displays the name of the Collapsed Statement or Collapsed Batch.
In MS-SQL	Displays entity resource consumption (default: stacked bar chart) and shows entity breakdown, over the selected time period.
%	Displays the percentage of the total In MS-SQL.
Occurrences	Number of different statements possessing the selected collapsed form.
Statement Executions	Number of statements executed within the selected time period by the selected entity.
Statement Duration (Avg)	Average duration for a single statement, in the selected entity, during the selected time period, calculated by dividing the Statement Duration by the number of executions.
Text	Displays the statement or batch text (formatted for easy readability), in the collapsed form.

About viewing In MS-SQL data for Collapsed Statement and Collapsed Batch entities

This tab contains the In MS-SQL counter and the breakdown of the In MS-SQL into its states. The In MS-SQL is displayed with a normalized view while all the other counters display the time each state was consumed.

About viewing Statistics data for Collapsed Statement and Collapsed Batch entities

The following table describes the additional information displayed in the Statistics tab in the Association area.

Table 10 Information displayed in the Statistics tab

Column	Description
Entity Name (such as Collapsed Statement)	Displays the name of the entity.
Database	Identifies the database on which the collapsed statement or batch is run.
CPU Time	Cumulative CPU time for the entity during the selected time period.
Physical I/O Operations	Total number of physical I/O requests performed by the selected entity over the selected time period.
Parallel Sessions (Min)	Minimum number of threads used to execute the statement or batch in parallel. This counter is calculated from the e `cid` column in `sysprocesses`. This enables you to verify that SQL Server is using the best execution plan for the current statement.
Parallel Sessions (Max)	Maximum number of threads used to execute the statement or batch in parallel. This counter is calculated from the e `cid` column in `sysprocesses`. This enables you to verify that SQL Server is using the best execution plan for the current statement.

About viewing Procedure Cache data for Collapsed Statement and Collapsed Batch entities

The Procedure Cache is the part of the SQL Server memory pool that stores execution plans for batches, stored procedures and triggers. Detailed information on the execution plan run of a specific instance is displayed in the Procedure Cache tab.

The following table describes the information displayed for execution plans in the Procedure Cache tab.

Table 11 Information displayed in the Procedure Cache tab

Column	Description
Entity name (such as Instance, Database)	Displays the name of the entity.
Elapsed Time	Amount of time it took to complete the executions of the plan.
Execution Count	Number of times the plan was executed.
Avg Duration	Average amount of time it took to execute the plan, calculated as total elapsed time divided by the execution count.
Recompilations	Number of times a plan was recompiled while it remained in the procedure cache.
CPU Time	Amount of CPU time consumed by the executions of the plan.
CLR Time	Time consumed inside CLR objects by the executions of the plan.
Logical I/O	Number of logical reads and writes performed by the executions of the plan.
Physical Reads	Number of physical reads performed by the executions of the plan.

About Session and Blocker Session entities

The Session entity displays performance, statistics and locking information on the selected session. This entity differs from other entities in the Activity tab in that session information is retrieved from the Collector's temporary files and not from the PMDB.

The Blocker Session entity displays information on the holding locks that cause other sessions to wait for resources to be free.

The following views are available:

- Overview
- In MS-SQL
- Locking
- Internal Waits
- Overall Activity

About getting an overview of Session and Blocker Session entities

The Overview displays general details on the selected session or blocker session, such as, Machine, Login, and Client Process ID. A breakdown of the selected session's or blocker session's resource consumption breakdown is also displayed.

About viewing In MS-SQL data for Session and Blocker Session entities

The In MS-SQL view displays the resource consumption of the selected session over the selected time period. This graph enables you to analyze performance trends for the session, over time.

> The graph displays data in time slices.

See About session states and Focusing on information in overtime graphs.

About viewing Session and Blocker Session Locking

The Locking view displays the blocking tree and lock wait duration broken down into lock levels (such as, Table, Page, and Key), for the selected session, within the specified time period. The following information is displayed:

- **Locks**. Displays which sessions were involved in a blocking situation (either locking or blocking). It is important to understand which sessions are involved in each blocking chain, as there may be several sessions involved at the same time. It is then important to determine which session is blocking others, which object is being blocked, and what type of lock has been applied.
  A yellow lock indicates that the session is currently blocking other sessions but is not itself blocked.
  A red lock indicates the session is currently blocked by another session. Note that a session may also block other sessions while being blocked by a different session.
- **Lock wait graph**. The Lock Wait graph shows a breakdown of the Lock Waits, and when they occurred, over the selected time period.

> The graph displays data in time slices.

About the lock tree structure

To gain a better understanding of the lock tree let's take a look at its structure. Notice that the session located in the root of each chain is the session that blocks all others. Each session directly blocks all sessions positioned one level beneath it.

The following information is displayed in the Lock tree:

- start time of the lock
- duration of the lock
- the SPID of the sessions involved in the lock
- the program name and the client process ID of the session.

Focusing on a blocked session branch in the lock tree will expose the SQL text performed by the session. The lock type is displayed in the Lock Breakdown table, which is displayed to the right of the Lock Trees.

Every time a change occurs, a new lock tree is opened, as for example, when a new session enters into the lock. It is therefore possible that there will be sub-trees of several locks that may appear identical. See Focusing on information in overtime graphs.

About viewing internal waits for session and blocker session entities

The Internal Waits view displays the following internal waits, indicating that the session is waiting for an internal resource to be freed:

- **Other internal wait**. Aggregates the following types of waits:
  - Full text waits
  - HTTP waits
  - Query notifications
- **Backup wait**. Includes the wait type that commonly occurs when a Backup command is performed.
- **Memory wait**. Includes ten types of waits that indicate that a session is waiting for memory to be allocated to it.
- **Profiler wait**. Aggregates a number of states associated with the SQL Profiler and lets you see how much of the database resources it consumes.
- **DB mirror wait**. Aggregates the new waits that occur when DB mirroring is performed, such as the waits that occur if the communication layer used by DB mirroring becomes backlogged.
- **DTC wait**. Aggregates waits that occur when Distributed Transaction Coordinator sessions have to wait for each other.
- **Parallel wait**. The session is waiting for one of its sub-threads to complete its operation.
- **Latch wait**. The session is waiting for an internal lock to be released.
- **Buffer pool wait**. Groups together the events that show contention on pages in the buffer pool.

About viewing overall activity for Session and Blocker Session entities

The Overall Activity view displays the duration of the activity performed by the selected session within the specified time period, broken down into the following components:

- **Waitfor Command**. The amount of time applications were waiting because they were waiting for command.
- **Parallel Exchange Wait**. The amount of time the thread in a parallel session is waiting for data exchange from another thread.
- **Request Wait**. The amount of time applications were in a request wait state, defined as the state wherein the server is waiting for a client request.

- **In MS-SQL**. Represents total amount of time spent in SQL Server.

> The graph displays data in time slices.

See Focusing on information in overtime graphs.

About the Session and Blocker Session entities Association area

The following tabs are displayed in the Association area of the Session and Blocker Session entities:

- Performance tab (Session and blocker session)
- In MS-SQL tab (Session and blocker session)
- Statistics tab (Session and blocker session)

Clicking a tab shows additional information on the entities displayed in the Association area table. The rows of the table remain the same; the columns change to display additional details on the entities listed in the table. The information displayed in each tab is described below.

About viewing Performance data for Session and Blocker Session entities

The following table describes the additional information displayed in the Performance tab in the Association area.

Table 12 Information displayed in the Performance tab

Column	Description
SPID	Displays the system process ID. The SPID is a unique integer assigned to each user connection when the connection is made.
Program	Displays the program that opened the session.
In MS-SQL	Displays entity resource consumption (default: stacked bar chart) and shows entity breakdown, over the specified time period.
Database	Displays the database used by session.
Login	Displays the SQL Server or Windows login that opened the session.
Start At	Displays the session's login time.
Completed At	Displays when the session was completed, within the selected time period.
Machine	Displays the machine the program ran on.
Work Type	Displays the group the session is related to, such as User Application or SAP Dialog.
User	Displays the database user that opened the session.
Duration (Summed)	Displays the time that has elapsed since the login time.
Client Process ID	Displays the process ID of the program.

About viewing In MS-SQL data for Session and Blocker Session entities

This tab contains the In MS-SQL counter and the breakdown of the In MS-SQL into its states. The In MS-SQL is displayed with a normalized view while all the other counters display the time each state was consumed.

About viewing Statistics data for Session and Blocker Session entities

The following table describes the additional information displayed in the Statistics tab in the Association area.

Table 13 Information displayed in the Statistics tab

Column	Description
SPID	Displays the system process ID. The SPID is a unique integer assigned to each user connection when the connection is made.
CPU Time	Cumulative CPU time for the entity over the selected time period.
Physical I/O Operations	Total number of physical I/O requests performed by the selected session over the selected time period.
Memory Usage	Displays the memory consumed (in MB) by the selected session over the selected time period.
Open Transactions	Displays the number of transactions that were opened by the selected session over the selected time period.

See About session states.

About the Locked Object entity

The Locked Object displays the time each object was locked. The collector determines that an object is locked according to its lock type and lock resources. The following lock types are displayed:

Table 14 Lock types

Type	Description
Database	Lock on the entire database.
Row In Table	Lock on rows in the table.
Key In Index	Lock on rows within the index.
Page of Database File	Lock on pages in the database file.
Extent of Database File	Lock on extents in the database file.
Table	Lock on the entire table's data and indexes.
Application	Lock on application resource.
Compile of Stored Procedure	Lock on stored procedure.
Compile of Trigger	Lock on stored trigger.
Compile of Function	Lock on stored function.
Index Database File	Lock on the entire index.
Unknown	Precise for SQL Server is not able to identify the lock type.

If the Collector identifies a blocking situation but is not able to identify the lock type and lock resource, then the sum of the wait on all locked objects will not be identical to the lock wait of the parent.

About getting an overview of a Locked Object

The Locked Object view displays the locked Object Name, Lock Type, and the Lock Wait Duration graph, over the selected time period, as follows:

- **Properties**. Displays the locked object's full name and lock type. For example, if the locked object's type is Table, the name will also include the database and owner of the table.
- **Lock Wait**. Displays the lock wait duration of the selected locked object, over the selected time period.

About Table and Index entities

The Table and Index entities display performance information regarding the selected table or index. Object performance is measured by how much the object contributes to the performance of the statements accessing the object. The contribution is calculated by multiplying the cost of the operation(s) that access the object (in percentage) with the In MS-SQL time of the statement. The sum of the results of all statements accessing the object represents the performance of the object. The I/O wait time of an object is calculated in the same way.

The following views are available:

- Overview
- Performance

About getting an overview of Table and Index entities

The Overview displays general details regarding the Table or Index such as which database the table exists in, who is its owner and the amount of time it spent in SQL Server, broken down into its index and table components, as follows:

- **Properties**. Displays the following table or index identifiers: Database, Table Owner, and Table name, and index name (if the entity is an index).
- **In MS-SQL access breakdown**. Displays the estimated resource consumption resulting from accessing the table and its indexes in a Table entity and estimated resource consumption resulted from accessing the index in an Index entity.

About viewing Table and Index performance

The Performance view displays the estimated In MS-SQL duration versus the I/O waits for the selected Table or Index over the selected time period, as follows:

- **In MS-SQL**. This graph displays the estimated In MS-SQL for the Table or Index over the selected time period. When the selected entity is a table, the graph shows the breakdown of both the Table and index In MS-SQL duration. When the selected entity is an Index, only the portion for the selected index is displayed.
- **I/O**. This graph displays the estimated I/O wait duration resulting from accessing the Table or Index over the selected time period. When the selected entity is a table, the graph shows the breakdown of both the Table and index I/O wait duration. When the selected entity is an Index, only the information for the selected index is displayed.

About the I/O Entity

The I/O entity displays the I/O wait duration logged for the selected entity. The I/O entity can be a logical file or a storage device.

About the I/O entity Overview view

This view displays general information regarding the I/O entity. For example, for a Logical file entity the information displayed is: Logical File, File Database and Physical File name (including the path). In addition, the I/O wait duration over the selected time period is displayed.

How the Activity tab can help you identify performance problems

You can identify a performance problem by doing one or more of the following:

- Examining resource consumption of an entire instance and database
- Identifying heavy resource consumers
- Examining resource consumption over time
- Examining I/O waits of different storage devices connected to a SQL Server instance
- Correlating SQL Server data with ERP/COM+ data
- Examining locking over time of a database entity
- Identifying offending SQL statements
- Identifying resource-consuming collapse statements or batches
- Examining the availability of an instance and database
- Examining the scalability of your application or SQL servers
- Identifying tempdb contentions

Examining resource consumption of an entire instance and database

When performing a tuning audit it is very important to analyze and understand the performance behavior of your instances and databases. You may have already drilled down to analyze the performance of a single query or program. However, examining the entire instance's behavior can alert you to the health of your system. The Instance (or Database) represents the performance of the average application. This helps you answer questions such as: "This query suffers from Log wait, but does the entire system suffer from lock wait?" If, for example, you discover that your system suffers from I/O wait and you tune a query that performs memory operations, you most likely will not improve the entire system's I/O wait problem. So examining the entire instance can provide a quick overview of the dominant resources consumed.

To examine the resource consumption of an entire instance and database

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. On the View controls in the Main area, click **Overview**. Examine the entire instance and determine which are the dominant resources that are consumed by your system.
4. On the Association controls in the Association area, click **Databases**.
5. On the Performance tab, place the cursor in the In MS-SQL column of the top database and view the information displayed on its resource consumption.

As a rule of thumb, a healthy system should have a high Using CPU value, a 10–15% I/O Wait, a Log Wait based on the nature of the application, and the remaining states should show values that are as minimum as possible.

Identifying heavy resource consumers

Precise for SQL Server enables you to drill down to application components (such as Logins, Databases, Programs, Machines, and Statements) to determine which application component consumes the most resources. This is accomplished by clicking on an application component in the Association area. This process is iterative and you may continue to drill down until you discover the application component that you want to tune.

There are several reasons to drill down and focus on different application components:

- During the tuning process, you drill down to determine which are the top resource-consuming components. Tuning a component (such as a Database, Login, User, or Machine) that consumes most of the resources, will free them to other entities. The tuning process is an iterative process. You will continue to list all components, examine their resource consumption and focus in on several of them, until you have completed the tuning process.
- When you try to analyze user experience you need to isolate one user's activity from the other users. To achieve this you can focus in on a single login or machine.
- When trying to determine which application component to drill down to, take into consideration not only the total In MS-SQL of the application, but also its number of sessions, executions and average duration. If you do not take these elements into consideration you may try to tune a program that runs once a year, consuming 20 hours, instead of tuning a program that consumes 2 seconds but runs every time a user logs on to the system.

To identify heavy resource consumers

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. On the View controls in the Main area, click **In MS-SQL** to view a breakdown of the In MS-SQL time.

> One of the components of In MS-SQL time that you can observe in the In MS-SQL graph is Internal Wait. You can view additional information on that particular component, over the selected time frame, by clicking **Internal Waits** on the View controls in the Main area. This view lets you analyze a breakdown of the Internal Wait parameter and determine which internal wait parameter is consuming the most time and resources.

4. On the Association controls in the Association area, click **Programs**.
5. On the Performance tab, place the cursor in the In MS-SQL column of the top program entity and view the information displayed regarding its resource consumption.
6. Drill down on the entity to focus on additional components.
7. Go back to the overview of the instance you originally selected.
8. In the Association area, use the Association controls to select different entities associated with the instance, and analyze the average duration of each entity.

Examining resource consumption over time

When you analyze and tune the performance of an application component (such as, database, program, or query) it is important to take into consideration its performance over time. Precise for SQL Server allows you to easily view the component's performance over a selected time period.

There are several reasons to study a component's performance over time:

- When analyzing a performance problem you will want to determine if it is a one-time only problem or has happened before.
- You will want to identify if there are any patterns or trends. For example, does a particular lock problem always happen on Tuesday at around 10 AM?
- You will want to determine how your system is behaving throughout an entire day. For example you may observe that during the night your system experiences much more I/O wait than during the day.

To examine resource consumption over time

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. On the View controls in the Main area, click **In MS-SQL**.
4. Move the cursor over the bars in the graph to view the instance's resource consumption over time.

Examining I/O waits of different storage devices connected to an SQL Server instance

If you have determined that the application is suffering from I/O waits, you may want to examine how the I/O waits are divided amongst the various storage devices. You will want to balance the I/O waits among all the storage devices.

To examine I/O waits of different storage devices connected to an SQL Server instance

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. On the View controls in the Main area, click **In MS-SQL**.
4. On the Association controls in the Association area, click **Storage Devices**.
5. Observe the I/O waits for each storage device unit.
6. If you locate storage devices suffering from I/O waits, you can check which logical files reside on the storage device and move a few to a different storage device. Drill down from the storage device and analyze the Logical Files entities. Once you have identified which logical file(s) should be moved to a different storage device, check the I/O pattern on that storage device, to ensure that the move will not cause the I/O pattern on the new storage device to deteriorate.

Correlating SQL Server data with ERP/COM+ data

Certain obstacles make it difficult to investigate ERP data in the SQL Server Tier. Most activity is registered in SQL Server under generic names. For example, in SAP, all activities are listed under a single login (SAP\SAPServiceSQ1) and some generic program names (R3D01(1)sp crea OLEDB).

Installing the Precise for SQL Server Interpoint extension lets you correlate SQL Server information with ERP information. For example, if you install Interpoint for SAP, you will be able to display the breakdown of your SQL Server activity into entities such as, SAP User and SAP Transactions.

The Precise for SQL Server with SAP Interpoint extension must be installed before you can correlate SQL Server data and ERP/COM+ data. For information on how to install the Precise for SQL Server with an SAP Interpoint extension, see the *Precise Installation Guide*.

The following table shows the benefits of installing the Precise for SQL Server Interpoint extension.

Table 15 Benefits of the Precise for SQL Server Interpoint extension

Type of Application	Benefit
SAP	Drill down to the most resource-consuming SAP transaction, SAP user, SAP work type, and SAP application server name.
PeopleSoft	Identify the most resource-consuming PeopleSoft Client Machines, PeopleSoft User IDs, Tuxedo OS Users, Tuxedo domains, and work types.

Siebel	Identify the most resource-consuming Siebel business object, Siebel application, Siebel component, and Siebel server.
COM+	Determine the most resource-consuming COM+ application, COM+ component, COM+ method, and COM+ interface.

To correlate SQL Server data with ERP/COM+ data

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. On the Association controls in the Association area, click **Siebel View/Programs**.
4. Drill down to the entity whose data you want to correlate.
5. On the View controls in the Main area, click **In MS-SQL**.
6. Move the cursor over the instance heading to correlate SQL Server information with ERP information

Examining locking over time of a database entity

While many think that locking is bad, the truth is that locking is good—it maintains data integrity. The situations to avoid are contentions. Contentions are situations in which one application waits for a resource that is being held by another application.

Locking can become problematic if the portion of time the application is blocked lasts too long. If you identify a lock wait time that seems too long, further investigation is required. The Lock wait time is reported as part of the In MS-SQL breakdown.

Analyzing the blocking situation is much easier when the lock is currently active, but this is usually not the case. For this reason it is important to understand why the locks occurred in the past.

There are several areas you can focus on when resolving a blocking situation:

- Look at the Overview view of your instance (database, program) to understand the percentage of Lock Wait your application suffers.
- Examine the Lock Wait over a selected time period. By examining resource consumption over time you can identify if Lock Wait occurred all the time, or only on specific dates.
- Try to gain an understanding of the Lock types your applications were blocked on. An overtime graph of the different Lock types can be displayed in the Locking view.
- Examine the locked objects. You can display a list of all locked objects and compare them according to lock time.
- Once you have identified all of the above, you also need to assess who is blocking who. For example, you will need to determine which session is blocking which session. The locks tree displays all the contentions that occurred and shows which sessions were involved in contentions.

To examine locking over time of a database entity

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. In the Association area drill down to the database entity you want to analyze.
4. On the View controls in the Main area, click **Locking**.
5. Examine the locking of the selected entity in the graph, for the period of time you selected. Move the cursor over the bars in the graph to view the Lock Wait breakdown (such as, table locks, row locks, page locks, key locks and other, in addition to total Lock Wait time for the selected time slice). Examine the locks tree to observe all the contentions that occurred and determine sessions were involved in contentions.
6. On the Association controls in the Association area, click **Locked Objects**.

Identifying offending SQL statements

Having identified a major resource-consuming entity, you can further drill down to the statements and batches executed by that entity. These may include long running SQL statements and SQL statements that use few resources but were executed frequently.

In many cases analyzing the queries executed by the entity and tuning them will result in the most performance gain. For each statement, you can do the following:

- View which entities executed the statement.
- View resources consumed by the statement, which enables you to determine if the statement is CPU or I/O bound, waiting for a lock, etc.
- View resource consumption patterns, which allow you to observe if the statement execution is affected by excessive use of the same resource by another statement.
- Tune the statement in the SQL tab.

To identify offending SQL Statements

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. In the Association area, drill down to the top resource-consuming database.
4. On the Association controls in the Association area, click **Statements**. On the Performance tab view which entities executed the statement and view the resources consumed by the statement.
5. On the Plan tab, in the Association area, view a summary of the statement's access plan.

Identifying resource-consuming collapse statements or batches

Many statements may look the same but differ in their literals. When you want to identify the most resource-consuming statements you would want to treat them as the same statement. Precise for SQL Server lets you identify the most resource-consuming collapsed statements or batches.

Once you focus on a collapsed statement you can examine the different statements having the selected collapsed form to examine the performance of each individual instance.

If you observe different performance patterns for the same statements you can also examine the distribution of their execution plans. If they have different execution plans this may explain the different performance patterns.

> The batch hash value for a collapsed batch in a stored procedure is calculated according to the header or definition. This lets you can observe the performance of a batch over time, even if changes were made to a statement in the batch.

To identify resource-consuming collapse statements or batches

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. On the Association controls in the Association area, click **Statements (Collapsed)**.
4. Drill down to the top resource-consuming collapsed statement.
5. On the Performance tab in the Association area, examine the different statements having the selected collapsed form to examine the performance of each individual instance.
6. On the Association controls in the Association area, click **Plans** to analyze the various access plans of the collapsed statement you selected.

Examining the availability of an instance and database

Because Precise for SQL Server maintains a constant connection with your SQL Server instance, it can easily report on the availability of the instance and databases.

The availability report displays the percentage of time the instance was up and the amount of time the instance was unavailable, over the selected time period.

> This view is accurate only if the Precise for SQL Server collector is up and running.

To examine the availability of an instance and database

1. In the Instance list, choose the instance you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. On the View controls in the Main area, click **Availability**.
4. Move the cursor over the Availability graphs and Unavailability Times graphs to observe how often the instance was available during the selected time period.

Examining the scalability of your application or SQL servers

Applications are called upon to support the addition of more and more users, and data, over the years. One of today's tasks is ensure that our application and servers are scalable for tomorrow's tasks. Precise for SQL Server enables you to examine if the application is scalable, or if the SQL servers are scalable.

To analyze scalability of an application you need to compare the number of sessions and the impact of the average session duration. In a well scalable instance, growth in the number of sessions will not impact the average session duration. This can be observed in the Scalability View in the Session Duration (Avg) vs. Sessions graph. The same applies to I/O and CPU, as you can see in the Physical I/O Operations vs. I/O Wait graph, which shows how more and more I/O requests affect the I/O waits.

When examining if the server is scalable, check if SQL Server CPU consumption growth affects the server CPU consumption and I/O wait times. You will need to answer if growth in the I/O requests increases the time I/Os are waiting in the Queue to be served.

The first example below shows that the application is not scalable. The Session Duration (Avg) decreases when the number of sessions increase.

To examine the scalability of your application or SQL servers

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.
3. On the View controls in the Main area, click **Scalability**.
4. In the Session Duration (Avg) vs. Sessions graph, observe the number of sessions and the impact of the average session duration, over the selected time period. For example, if you observe that average session duration increases when the number of sessions decrease, this may indicate that the application has a scalability problem.
5. On the View controls in the Main area, click **System Scalability**.
6. Analyze the In MS-SQL vs. SQL Server CPU usage graph to determine if SQL Server CPU consumption growth affects the server CPU consumption and I/O wait times. Check the Physical I/O Operations vs. Queue Length graph to determine if a growth in I/O requests increases the time I/Os have to wait in the queue to be served.

Identifying tempdb contentions

Tempdb is the most heavily shared resource across the entire instance and can affect the performance of all applications. It is therefore important to monitor its usage. A high tempdb wait value may indicate a bottleneck in tempdb.

To identify tempdb contentions

1. In the Instance list, choose the item you want to analyze.
2. In the Time Frame list, choose the period of time you want to analyze.

3. On the View controls in the Main area, click **In MS-SQL** to view a breakdown of the In MS-SQL time. Check the **Tempdb Wait Time** component of the In MS-SQL counter.
4. On the Association controls in the Association area, click **Programs**.
5. On the Performance tab, place the cursor in the In MS-SQL column of the top program entity and check which portion of resource consumption can be attributed to tempdb wait.
6. Drill down on the entity to focus on additional components.