Precise for SQL Server

User Guide

Version 10.2

IDERA

Precise for SQL Server User Guide

Global Headquarters
Brookhollow Central III
2950 North Loop Freeway West, Suite 700
Houston, Texas 77092
Phone: +1-877-693-1886
Fax: +1-650-898-1666

Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Precise product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

Customer Support

For technical assistance, visit our customer portal at http://www.idera.com/support/productsupport where you can find an extensive knowledge base, product updates and our online community forums. You can also contact our Customer Support Team using our customer portal, or go to http://www.idera.com/support/productsupport for a list of our support access numbers in your country.

Contents

Introducing Precise for SQL Server

This section includes the following topics:

- [About Precise for SQL Server](#)
- [Components of Precise for SQL Server](#)
- [How Precise for SQL Server works](#)
- [What you can do with Precise for SQL Server](#)
- [Where to get more information](#)

About Precise for SQL Server

Addressing the needs of DBAs, system administrators, managers, quality assurance specialists and planners, Precise for SQL Server provides a complete view of application performance by capturing, measuring, and correlating performance metrics from all critical system components.

Through its wide range of data presentations, Precise for SQL Server provides the tools necessary for proactive monitoring, diagnosing, detecting, and correcting of performance problems before they become visible in your SQL Server instances and related applications.

Current operational data is displayed to enable easy identification of problematic conditions. Recent operational data, pertaining to designated time periods, enables operations staff to detect problematic conditions that occurred in the past and that are likely to reoccur if not corrected. Performance data is archived and a fully detailed display of the historical data is provided for use in performance, QA and capacity planning analysis.

Precise for SQL Server's SmarTune engine provides information on performance findings that may indicate potential problems in the database. SmarTune displays a list of statements that may require tuning, a list of objects experiencing heavy activity or possessing an inefficient structure, and a list of instance events. Built in to the SmarTune engine, is a wealth of expert knowledge regarding possible areas of deterioration, and links that can guide you to the resolution of specific problems that will enable you to optimize the performance of your system.

Improved performance management

IT staff frequently search for answers to questions like "Why did the application run so slowly last night?" or "Where is the application spending its time?"

Precise for SQL Server can help IT staff discover the answer to questions, such as, why an application ran so slowly during a particular time interval, or what resource is being used most of the time. The first step is to establish a baseline level of database resource consumption by understanding the behavior of the average application. The behavior of the average application represents the average percentage of the total time that all applications spend consuming or waiting for each resource. These aggregate figures provide a broad picture of the operation of your instance. They enable you to identify major bottlenecks in your system, for example, CPU and I/O limitations, SQL Server instance waits due to locks, remote queries, and so on.

Precise for SQL Server also enables you to focus on and drill down into individual applications, SQL statements and database objects in your system that are responsible for heavy consumption of system and database resources and to quickly find the definitive root cause of performance degradation.

Integration with other Precise products

Precise for SQL Server has been integrated with the following Precise products:

Insight	Insight monitors the response times of systems and breaks these times down into the various Tiers and components. Insight also provides tier-specific metrics to help understand the activities in each tier.
	For more information, see the *Insight User's Guide or Online Help.*
Insight SmartLink	Insight SmartLink pattern recognition engine correlates end-to-end activities and provides information on the application's performance, from the user's perspective.
	For more information, see the *Insight User's Guide or Online Help.*
Alerts	Alerts provides a notification system that will alert you to application response time problems and enables you to launch Precise for SQL Server in context to view more detailed data.
	For more information, see the *Alerts User's Guide or Online Help.*
Report Manager	Report Manager gathers and organizes historical information that enables the IT team to focus on infrastructure hot spots. Report Manager is a reporting tool that resides on a Web server. You can use Report Manager to compare period-to-period performance against a baseline and identify response time problems "at-a-glance", before they affect your bottom-line. It queries the PMDB and displays the results in reports that are generated on a scheduled basis or on demand.
	For more information, see the *Report Manager User's Guide* or *Online Help*.
Precise Custom Portal	The Precise Custom Portal is a lightweight, configurable portal application that provides access to various types of data from different applications. The Precise Custom Portal features a Web-based view. It is highly customizable and extensible, allowing you to build a dashboard for each user or each function within your organization. For example, you can build a dashboard for all the information that a system administrator or director of IT operations would need to constantly follow up on. This component is automatically installed as part of the Precise framework installation.
	For more information, see the *Precise Custom Portal User's Guide*.
Precise for SAP	Precise for SAP offers a wide range of data presentations that provides you with the information necessary to manage SAP client/server systems. Precise for SAP analyzes SAP system performance and applications and helps you to effectively direct resources and enhance system use.
	For more information, see the *Precise for SAP User's Guide* or *Online Help*.
Precise for Microsoft .NET	Precise for Microsoft® .NET, is a comprehensive performance management product for .NET applications that addresses the needs of system operators, managers, quality assurance specialists, and planners. It provides the specialized data collection and analysis capabilities demanded by the critical role of the .NET in application performance and scalability.
	For more information, see the *Precise for Microsoft .NET User's Guide* or *Online Help*.
PMDB	The PMDB is the repository of the Precise for SQL Server suite which stores application performance data for long-term analysis and trending reports. The data is collected by the Precise product suite.
	For more information, see the *Precise Administration Guide* or *AdminPoint Online Help*.

Components of Precise for SQL Server

Precise for SQL Server employs a web-based architecture to provide high performance database monitoring and tuning.

Precise for SQL Server consists of the following components:

- One or more Precise for SQL Server Collectors installed on one or more servers in your application.

- Precise for SQL Server FocalPoint

- PMDB

- Collector extension, such as storage Collector or ERP Collector.

For more information, see the relevant product's manual.

The following figure illustrates the typical architecture of a Precise for SQL Server installation.

Figure 1-1   Precise for SQL Server architecture
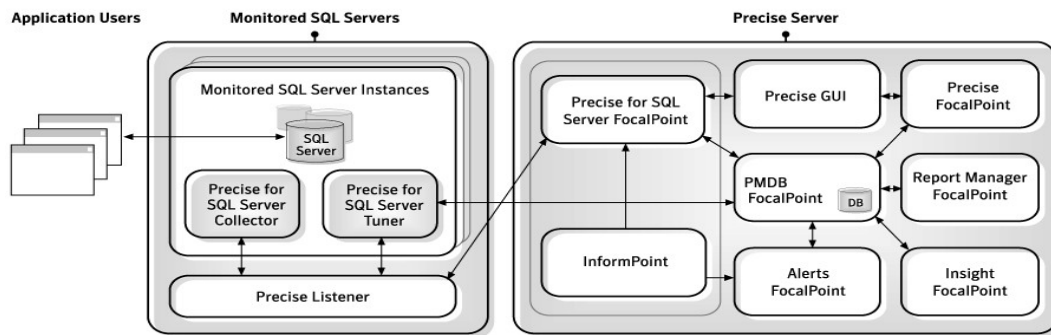
Precise for SQL Server agents

The Precise for SQL Server Collectors are installed on servers and communicate using TCP/IP. The Collectors collect and analyze SQL Server instance performance data, which is then stored in flat historical files and later in the PMDB.

The Collectors monitor and record all SQL Server database activity on a 24 x 7 basis. The Collectors work using a sampling technique and thereby consume less resources than a trace agent.

The Collectors collect resource consumption data on the following SQL Server data entities: databases, sessions, programs, statements, batches, login names, DB users, machines, SQL Server jobs and steps.

The following agents are available:

Application Performance agent	Precise for SQL Server provides a complete view of application performance management by capturing, measuring and correlating performance metrics from all critical system components.
SAP™, PeopleSoft and Siebel Collectors	If ERP support is installed, the Collector also includes the Interpoint for PeopleSoft, SAP or Siebel agents. This agent provides data on the activities and resource consumption of packaged application entities. This data can help you identify ERP users, transactions, reports, and so on, that are heavy users of SQL Server resources. The current version only supports SAP, PeopleSoft, and Siebel Collectors.
	COM+ Collector If COM+ support is installed, the COM+ Collector provides data on COM+ applications that originate in SQL Server activity. It can help you correlate your SQL Server instance and COM+ activity.
Storage agents	EMC Storage, HP Storage, Hitachi Storage, and Storage for Windows. See Storage systems.

Precise Listener and Precise for SQL Server FocalPoint

Each Precise product has a FocalPoint that communicates with the listeners installed on the monitored servers. The Precise Listener communicates with other Precise components on the same server. For example, the PMDB FocalPoint requests to load data into the PMDB are sent to the Precise Listener. Then, the Precise Listener executes the PMDB loader to aggregate and prepare performance data to be loaded. In a similar way, InformPoints send requests to the Precise Listener.

PMDB Loaders

PMDB Loaders periodically load data—collected by the Collectors and stored in the historical files—into the PMDB. When the PMDB Loaders run, the flat historical files—written after the last time data was loaded into the PMDB—are identified and opened. The summarized data is also loaded into the PMDB. Data on the last incomplete time slice is only loaded into the PMDB when the next scheduled load process runs.

Storage systems

The storage systems supported by Precise for SQL Server include EMC Storage systems, HP storage systems, Hitachi storage, Storage for Windows, and other non-EMC storage systems:

EMC, Hitachi, HP Storage systems, and Storage for Windows systems:	The EMC Storage systems supported by Precise for SQL Server include Symmetrix® units and CLARiiON® units. In addition, Precise for SQL Server supports HP storage systems, Hitachi storage systems, and Storage for Windows. For those storage systems, I/O wait mapping extends into the storage units, which means that you can see how the disks in these storage units are actually allocated and used.
	Precise for SQL Server displays information that relates the I/O wait to EMC Storage devices in which they are stored. This allows you to locate specific EMC Storage devices where performance problems are occurring.
	By analyzing the information collected from your SQL Server instances, Precise for SQL Server provides an accurate picture of I/O activity and all the information needed to tune the I/O system. Using data collected by the EMC Storage agent, you can determine the cause of problems in your EMC Storage unit and optimize data distribution to reduce I/O wait time and maintain instance performance levels.
	By identifying overactive EMC Storage devices, you can maintain performance levels by reorganizing storage configurations, reallocating your database objects to other file groups, or moving file groups to alternative devices. EMC Storage information can be viewed in the Activity and Objects tabs.
Other storage systems:	Other storage systems supported by Precise for SQL Server include storage systems that work with Microsoft® SQL Server™ 2000 and Microsoft® SQL Server™ 2005, running on Microsoft Windows® 2000 and Microsoft Windows® 2003 servers. For these storage systems, I/O wait mapping does not extend into the storage units; it ends at the host data level.
	For example, if you have a Network Appliance™ storage system installed, you can see the I/O wait on LUNs (Logical Unit Number), which are the physical disks in a Microsoft Windows® operating system, but you cannot see how these disks are actually allocated and used.

How Precise for SQL Server works

Precise for SQL Server stores a complete history of your SQL Server instance activity in the PMDB. This provides you with a performance tracking tool that gives you the ability to assess SQL Server instance performance trends and analyze changes in SQL Server instance activities over long periods of time. This data is vital for identifying patterns in resource consumption and predicting future resource consumption in your SQL Server instance.

The Precise for SQL Server Collector runs on your server using a unique, accurate and comprehensive sampling technology that requires low overhead. It collects performance data from an instance and then correlates and loads the data into the PMDB tables. This data is then analyzed and summarized by the FocalPoint. You can later access this data using the Precise for SQL Server Web browser interface.

The SmarTune process detects performance deterioration by regularly analyzing SQL Server performance data, and identifying less than optimal behavior.

What you can do with Precise for SQL Server

Use Precise for SQL Server to detect, diagnose and resolve performance problems in your SQL Server instance. Precise for SQL Server enables you to identify real-time performance problems (for example, locking problems) and monitor a complete history of your SQL Server instance.

You can use Precise for SQL Server to:

- Resolve performance problems by identifying their root cause and improving statement performance.
- Proactively monitor, analyze, and tune SQL Server instances.
- Analyze and tune locking problems.

- Determine whether an I/O problem is caused by a SQL Server application.

- Tune a database object rather than a specific statement that accesses the database object.

- Perform I/O tuning of storage devices.

- Identify the most resource-consuming SQL statement and view a detailed explanation of the execution plan of the SQL statement.

- Monitor your system's current state as well as previous states.

- Recommend which index would provide a better execution plan and reduce resource consumption. If you are running SQL Server 2005, you can also test and evaluate the cost of implementing statement tuning recommendations with the What-if option.
  The What-If tab shows the number of statements whose performance would improve and the number of statements whose performance would deteriorate by implementing the recommendation. It also shows the cost of implementing the recommendation so that you can weigh the pros and cons of implementing the change. Use the What-If option to evaluate a set of recommendations on the workload containing the most common Transact-SQL statements used by the application, during the selected time frame.

Where to get more information

More information on Precise, its products, technical notes, and so on, can be found in the Release Notes document for this version.

Precise for SQL Server basics

This section includes the following topics:

- About Precise for SQL Server tabs
- How most tabs are structured
- About drilling down in context
- About configuring your settings
- Tasks common to most tabs
- About role management in Precise for SQL Server
- Opening your Precise product from the Precise StartPoint

About Precise for SQL Server tabs

Precise for SQL Server continuously collects high-quality metrics from each supporting tier of your SQL Server-based custom applications infrastructure and correlates these metrics to build a clear picture of application performance from the end-user's perspective—that is, response time.

This information is displayed in the Precise for SQL Server tabs and provides you with the information necessary to successfully track your instance's performance and identify patterns in resource consumption.

The following table describes which tasks can be performed in each tab.

Table 2-1   Precise for SQL Server tabs

Tab	Description
Dashboard	Allows you to quickly identify which SQL Server instance has a performance problem, providing the first glimpse into where the problem lies.
Current	Allows you to solve the performance problems that are happening now.
Activity	Allows you to examine application performance over time and helps locate performance bottlenecks.
Objects	Allows you to identify which SQL Server objects are used inefficiently by the application, and their effect on the application components.
Statistics	Allows you to identify performance bottlenecks in the way SQL Server instance resources are used.
SQL	Allows you to tune problematic SQL statements or stored procedures.
Maintenance	Displays information on SQL Agent activities and other SQL Server activities related to DBA.

Table 2-1 Precise for SQL Server tabs

Tab	Description
SmarTune	Displays performance findings that may indicate potential problems in the instance. The list of problems includes the following: • List of statements that require tuning • List of objects displaying heavy activity or possessing an inefficient structure • Instance events

How most tabs are structured

Though each tab is structured differently, most tabs consist of two different areas. Each area can include different control elements, such as tabs and view controls, and displays information in various formats, such as tables, graphs, or charts. The various areas are related to each other in that performing an action in one area affects the information displayed in other areas on the page. For example, the Current tab contains an upper and a lower area. The lower area (Association area) shows information in a table format. Each row in the table represents an application. The upper area (Main area) displays general information on the selected entity as well as related applications.

The entities displayed in the Association area are associated with the selected entity or related application displayed in the Main area. At times, the relationship between the entity displayed in the Main area and those displayed in the Association area is that of parent to child , and sometimes it merely represents that there is a relationship between the selected entity and the entities displayed in the Association area (such as when a statement is the selected entity displayed in the Main area and the list of programs associated with the selected statement is displayed in the Association area).

It is possible to click a row and drill down to an additional level below the selected level or use the filter feature to determine which information is displayed in the table. 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) changes to reflect the action you requested.

You can control the tab by doing any of the following:

- Keeping the selected entity but changing the associated entities—by selecting a different entity from the Association controls in the Association area.

- Changing the selected entity—by selecting a row in the Association area.

- Changing the view for the selected entity—by selecting a different view in the Main area.

For example, when you want to view information on a specific application in the Current tab, choose Active Applications from the Association controls. The Association area changes to display applications-related information. The Tab heading and the Main area remain unchanged.

In the Association area, click the row of the application you want to view detailed information for. The Tab heading indicates the newly selected entity; the Main area displays information on the application you drilled down to, and the Association area shows information on statements executed by this application.

Figure 2-1 How a typical tab is structured
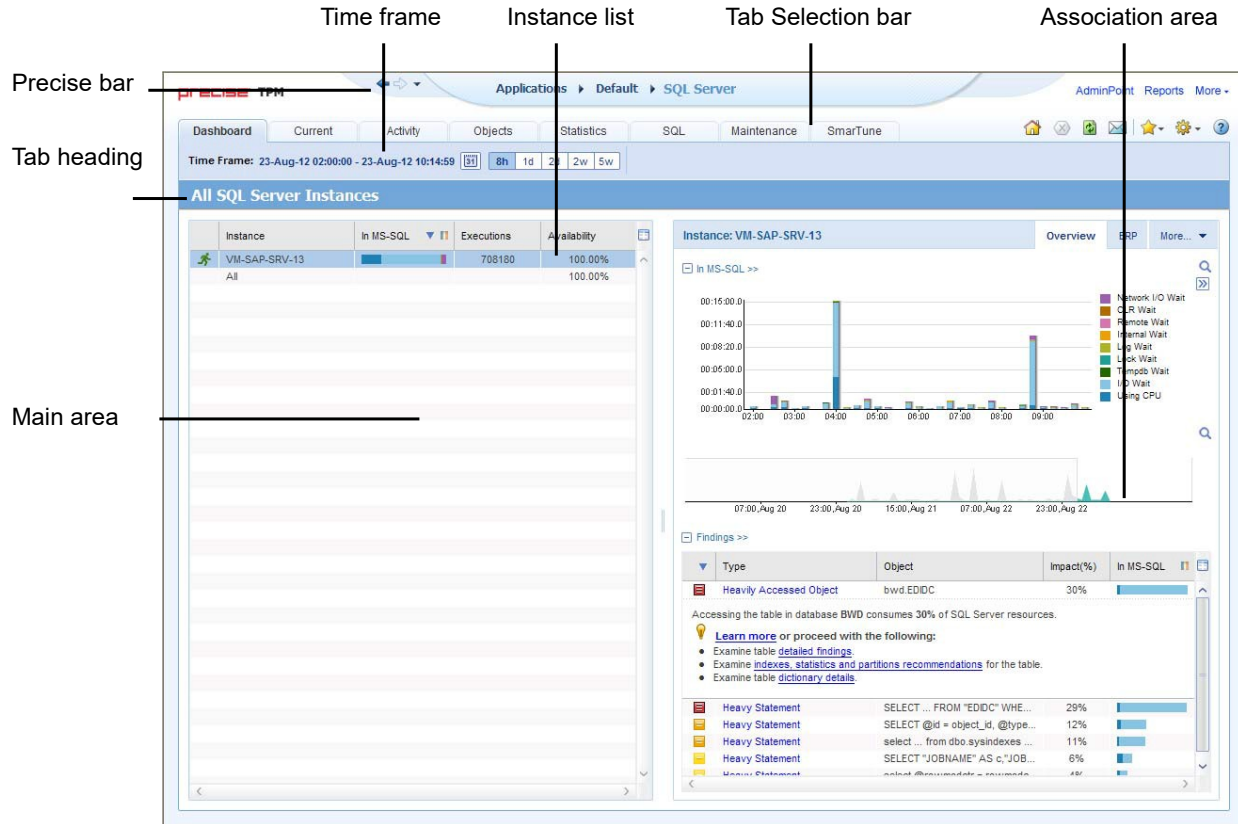
About the Precise bar

The Precise bar enables you to keep track of where you have been and provides various controls. The following table describes the function of each of the toolbar buttons.

Table 2-2 Precise bar functions

Icon	Name	Description
	Back	During a work session, keeps track of where you have navigated to. The Back and History buttons enable you to navigate between previously visited views. The Back control displays your previous view.
	Forward	Enables you to navigate to the next view. This button is only enabled if you clicked Back or if you chose a history option.
AdminPoint	AdminPoint	Opens Precise AdminPoint.
	Home	Navigates to the highest level entity, usually the instance or Tier (all instances). The time frame settings remain the same.

Table 2-2   Precise bar functions

Icon	Name	Description
	Stop	Stops a request for information from the server.
	Refresh	Updates the data currently displayed.
	Favorites	Enables you to add or remove favorites in your Favorites list.
	Send	Opens a new email message in your email program with the link to the current application in context.
	Settings	Opens the General Settings and Time Frame Settings dialog boxes.
	Help	Opens the online help in context.

About the Main area

The Main area displays general information on the selected entity. The structure of this area depends on the selected entity and tab.

Information can be displayed as follows:

- As two over-time graphs that are displayed side-by-side and describe two data series.
- As a pie chart and a table that are displayed side-by-side. The table displays more details regarding a specific slice in the pie chart.

The View Control drop-down menu allows you to choose different view options. You can access this menu by clicking the arrow icon.

In some tabs, you can filter the contents of the view by clicking the Filter button. See Filtering data.

About the Association area

The Association area lists entities that have similar attributes and that are related to the selected entity named in the Tab heading. Besides the Tab bar, the Association area is the main navigation tool. It enables you to drill down from one entity to related entities (usually of a different type).

The entities displayed in this area depend on the selected tab and the selected entity. The same entity may be displayed in several tabs, with different data displayed for each one (for example, a SQL table will be displayed in the Objects tab with data from the SQL data dictionary, and in the Activity tab with data sampled by the Collector).

The Association control, accessed by the arrow icon, allows you to associate with different entities. The last option, More..., opens a dialog box that lets you carry out additional operations, such as to view additional associations, change the sort order, or control the number of returned rows.

The information displayed in the Association area is arranged in a tabular format. Each row in the table represents an entity, and each column describes an attribute of the entity, such as text or a graph. It is possible to change the sort order of the entities in the table by clicking a column title.

The Association tabs located above the table let you display different sets of attributes for the entities displayed in the table. The Association tabs are entity-dependent. For some entities, no tabs are displayed.

A numeric column can optionally display a graph, and at times, it can display a stacked bar graph as well. Switching to a graph view in these columns offers some level of breakdown to the information displayed in the column.

The following icons are used to indicate which format information is displayed.

	Displays the information in numerical format.
	Displays the information in a bar graph.
	Displays the information as a stacked bar graph.
	Displays the information in percentage format

The Association area lets you drill down to another entity by clicking a row. The information displayed in both the Main area and the Association area will change to reflect your selection.

See Filtering data and Associating entities with data that meets specific criteria.

About the Findings area

For selected tabs in Precise products, the association area includes the Findings area, displaying problematic findings for the application. The findings feature is a high level tool, designed to provide the user with an overview of performance issues within the monitored application and enable quick and efficient navigation to the relevant tab for further analysis and handling. The displayed performance findings may indicate performance deteriorations as well as incorrect methods usage. Each finding appears as a row in the displayed table of findings. Hover the mouse indicator over the single-line displayed finding to expand the finding. When expanded, the finding details area provides important guidelines as to what may be the root cause for the reported problem, and what the recommended steps are to resolve this problem.

The table below describes the information that is displayed in the Findings area.

Table 2-3   Information displayed in the Findings area

Column	Description
Severity	The severity of the finding is calculated using a formula. The position of the finding in the list is determined by an internal scoring system that is based on the knowledge of Precise product experts. The severity is indicated by the following colors: • Red: High severity • Orange: Medium severity • Yellow: Low severity • Blue: No severity - the finding is strictly informative By default, findings are displayed according to severity.
Finding	A short name of the Finding.
Context	Entity/Method name (unless specified for the whole instance). The entity/method name is a short name but the long name is displayed in the ToolTip. Some of the findings are identified in specific Methods while others are relevant for the entire instance. In the latter case, a finding is specified as an instance-related finding. **Note:** Rows are aggregated by finding and the URI/method/SQL name and not by ID.
Finding overview	Displays specific details regarding the finding in context. **Note:** This is displayed in the expanded view only.

Table 2-3 Information displayed in the Findings area

Column	Description
Learn more (advice)	Provides recommendations for solving the selected finding. For each finding, it lists all relevant pieces of advice and all applicable solutions. You should carefully review all data for the finding and then choose the advice that best suits your needs. **Note:** This is displayed in the expanded view only.
Proceed with the following (bullets)	Provides expert knowledge about the selected finding. The information displayed will direct you if you have difficulties deciding which advice to take or which solution to implement. **Note:** This is displayed in the expanded view only.

Investigating a finding

Perform the following steps to investigate a finding. To investigate a finding

1. In the Time Frame list, select the period of time you want to analyze.

2. In the All Instances table, select the instance you want to investigate.

3. In the Finding area, review the top Findings for the selected instance displayed in the Findings table. The findings displayed in this table are sorted by severity.

4. In the Findings table, select the finding you want to analyze further.

5. In the selected finding (the expanded view), read the data displayed for the finding and follow any links provided to view additional information (advice) or next steps (bullets) to resolve the problem.

For more information, see SQL Server Findings.

About drilling down in context

The term 'in-context' means that you can display additional information on a selected item by drilling down to another tab or view. The filter settings you defined (for example, the selected time frame you chose) and the entity you selected are carried over to the other view or tab, to allow you to continue analyzing your subject from a different perspective. This concept takes on slightly different meanings depending upon where you are attempting to drill down in context from.

For example, when viewing information on a database partition in the Dashboard tab, you can click a link in the Details area (right pane) to view additional information on the related tab, in context of your original selection.

Alternatively, when viewing a list of alerts for your product in Alerts, you can launch your product in context and continue investigating the factors that led the system to issue that alert.

In short, the information displayed when drilling down in context is always related to your original selection's settings.

How a drilldown affects the Tab heading

The Tab heading displays the name of the currently selected entity on this screen. The Tab heading changes to reflect the name of the newly selected entity when you drill down to a new entity in the Association area.

See How most tabs are structured.

About configuring your settings

The following settings allow you to control the appearance and behavior of the user interface. They can be configured from the Settings menu on the Precise bar:

- General settings - Includes the General, Display, and Tree view tabs.

- SmarTune settings

- Time Frame settings

About configuring general settings

On the General tab of the General Settings dialog box, you can adjust the following:

- **Number of items that will be displayed for each association**. From the list, select the maximum number of items that will be displayed in a table. The default is 50 items. You can control the number of items returned for a specific association by using the More... option in the Association controls. This way, you can display up to 500 items.

- **Maximum number of characters to display in a text ToolTip**. When you move your cursor over text columns in a table listing Statements or PL/SQLs, a ToolTip appears, displaying more text than can normally fit into a table cell. This field controls the maximum number of characters that can be displayed. The default is 500 characters.

- **How often the information in the Current tab is refreshed**. If you select the Auto-refresh Current tab every check box, the Current tab automatically refreshes at the specified interval. By default, the tab is not refreshed automatically. If you select this option, the default refresh interval is every five minutes.

About configuring tuner settings

In the Tuner tab you can change the Login name and Password used when launching Microsoft® SQL Server Query Analyzer and the Recommend Indexes processes.

About configuring display settings

When you drill down to or choose a different entity, selecting the option on the Display tab of the General Settings dialog box allows you to maintain the selected view or tab when changing from one tab to another. This is sometimes referred to as sticky tabs.

About configuring Tree view settings

(Only applies to Precise products that include a Tree view)

On the Tree View tab of the General Settings dialog box, you can adjust the following:

- Number of categories to display when expanding an item.

- Number of items to display when expanding a category.

About configuring SmarTune settings

You control the maximum number of findings to display in the Findings area and whether or not you want to ignore findings, in the SmarTune settings dialog box.

See "Configuring Findings settings" on page 212.

About configuring Time Frame settings

You can determine the resolution of the data that is displayed in the overtime graphs using the Time Frame Settings dialog box. By using this dialog box you can define the default time frame to display.

Tasks common to most tabs

The following tasks are commonly performed in most tabs:

- [Switching to a different tab](#)

- [Selecting a time frame](#)

- [Selecting an entity](#)

- [Filtering data](#)

- [Associating entities with data that meets specific criteria](#)

- [Saving a statement](#)

- [Focusing on information in overtime graphs](#)
- [Sending an email message](#)
- [Adding, viewing, and deleting Favorites](#)
- [Determining which table columns to display](#)
- [Copying data to the clipboard](#)
- [Exporting to the Precise Custom Portal](#)

Switching to a different tab

You can easily switch between the different tabs using the Tab Selection bar. When you start your Precise product, the Dashboard tab opens by default. For other Precise products, another tab will open by default. The button of the selected tab is displayed in orange.

To select a tab, click a button on the Tab Selection bar to display information on the selected entity in a different tab.

Selecting a time frame

You can configure Precise for SQL Server to display transaction performance data for a specific time frame using the predefined time frame options or calendar icons.

Selecting a predefined time frame from the toolbar displays transaction performance data for the selected time period up to the current time.

See "Selecting a predefined time frame from the Precise for SQL Server toolbar" on page 29.

Selecting the time frame using the calendar icon, you can choose to define a time range independent of the current time, or to define a time range up to the current time.

See [Selecting a time frame using the calendar icon](#). The predefined time frame options are:

- Last 8 hours (8h) (default)
- Last 1 day (1d)
- Last 2 days (2d)
- Last 2 weeks (2w)
- Last 5 weeks (5w)

The time frame selected affects all information displayed in Precise for SQL Server. Only data that falls within the selected time frame is shown in these areas.

Selecting a predefined time frame from the Precise for SQL Server toolbar

To select a predefined time frame, from the Precise for SQL Server toolbar, select one of the predefined time frames.

Selecting a time frame using the calendar icon

To select a time frame

1   Click the calendar icon. In the dialog box that is displayed perform one of the following:

   a.   To define a time frame independent from the current time, select the 'Time Range' option and select the Start and End dates and times.

   b.   To define a time frame up to the current time, select the 'Last' option and enter the desired time frame.

   c.   To use one of the three previously used time frames, select the 'Recently used' option and from the drop-down menu select the desired time frame.

   d.   To use a previously saved time frame, select **Use a previously saved time frame** and from the drop-down menu select the desired time frame.

2.   To save your settings for future access, select **Save these definitions for future use as:** and enter a name in the corresponding field.

3.   Click **OK**.

Selecting an entity

You can select the entity that you want to examine in more detail. To select an entity

- On the Main area of the tab, select the entity you want to examine.

Filtering data

You can focus on specific contents of the Dashboard and Activity tabs by showing a subset of the information. This shows the contribution of entities such as programs and users.

You can define flexible selection criteria in the Filter dialog box. To open the dialog box, you need to click the Filter Off icon next to the Time Frame selector. Filter On or Filter Off indicates the current state of the filtering mechanism. Filter Off means all information is shown; Filter On means a subset of information is shown.

When you apply your selections in the Filter dialog box, the information displayed in both the Main area and the Association area is modified to reflect your selections. Also, the filtering continues to apply when you drill down to associated entities.

Note: You can enter multiple criteria, in which case each criterion is applied using the logical AND operator.

To filter data

1   Click **Filter is Off/Filter is On**.

2   In the Filter dialog box, do the following for each entity you want to filter:

From the left drop-down list	Select an entity.
From the middle drop-down list	Select an operator, such as, Like, <>, Not Like, In, Not In.
In the text box	Type the criteria (case-sensitive) for the selected entity. If you select the operator Like or Not Like, you can use the % wildcard character to represent 0 or more characters, and the _ wildcard character to represent exactly one character. If you select the operator In or Not In, type a comma to separate values.

3   Click **OK**.

Associating entities with data that meets specific criteria

You can associate the displayed entity with specific data to focus your analysis.

Note: The criteria no longer apply when you drill down to another entity.

To associate entities with data that meets specific criteria

1   Click the arrow located to the left of the Association controls and select **More...** (applies to Current and Objects tabs only)

2   In the Associate With dialog box, on the Entries tab, select the entity you want to associate data with from the Populate table with list.

3   In the Sort entries by list, determine according to which criteria you want the information to be sorted and in which order.

4   From the Display top list, select the number of rows to display.

5   On the Criteria tab, do the following for each entity you want to associate data with:

From the left drop-down list	Select an entity.
From the middle drop-down list	Select an operator, such as, Like, <>, Not Like, In, Not In.
In the text box	Type the criteria (case-sensitive) for the selected entity.
	If you select the operator Like or Not Like, you can use the % wildcard character to represent 0 or more characters, and the _ wildcard character to represent exactly one character. . If you select the operator In or Not In, type a comma to separate values.

6   Click **OK**.

Saving a statement

Save As allows you to save the statement or batch so that you can examine its access plan at a later date.

To save a statement

1   Select the statement you want to save in the Association area.

2   Click **Save As**.

3   In the Save Statement dialog box, enter the following properties:

Instance	The name of the instance that the statement belongs to.
Cabinet	The name of the cabinet that the statement is saved in (creates a cabinet if it does not already exist).
Folder	The name of the folder that the statement is saved in (creates a folder if it does not already exist).
Name	The name of the statement (generated randomly).
Database	The name of the database where the statement is running.
User	The statement's parsing user.

4   Enter the SQL text for the statement in the Text frame.

5   Click **OK**.

The new statement is saved in the PMDB in a logical cabinet and folder hierarchy.

Focusing on information in overtime graphs

Some entities display an overtime graph. The overtime graph displays entity statistics over a specified time period. Depending on the number of points displayed in the graph, you may need to zoom in or out. The text displayed on the x-axis varies according to the time frame. If there is a year or day change, x-axis labels will display accordingly.

To focus on information in overtime graphs, select the desired time frame on the overtime graph, click and drag the left or right handle to increase or decrease the time range. The small zoom (spyglass) icon will display on the upper right of the selected time range, and above the overtime graph legend. Click the zoom icon to zoom in according to the selected time frame.

Sending an email message

You can send an email message to one or more recipients from the Precise toolbar.

The default subject for the message will be "Link to a Precise application".

The email will include a link to the Precise product in the current context (time frame and selected entries).

To send an email message

1   Click the email icon on the Precise toolbar. The default email program opens.

2   Fill in the required fields and click **Send**.

Adding, viewing, and deleting Favorites

The Favorites feature allows you to save a specific location in your application and to retrieve the same location later without having to navigate to it.

About the Favorites feature

The new Favorites feature includes the following options:

- Relative Time Frame – Saving a relative time frame instead of static date. For example, saving the last 7 days will always display the last 7 days, depending on the day entered.

- One click to specific location – Once you open Precise by launching a saved Favorite item, you will not have to enter a login credential nor click the login button.

- IE Favorites support – Adding a new Favorite item in Precise will also add it to the IE Favorites menu.

- Auto Complete – The Favorites dialog includes a new combo box which supports AutoComplete.

- Auto Naming – The Favorites dialog generates item names based on the current location.

UI description

A Favorites menu has been added to the Precise bar in each product including StartPoint.

An Add/Delete Favorites option under the Favorites menu allows you to save the current location or delete an existing one.

To add a new Favorite location

1   On the Add/Delete Favorites dialog box, enter the name of the new Favorites entry.

2   Click Add. The dialog box is closed and the new Favorite is added to the list.

To view a Favorites location

1   On the Precise bar, click **Favorites**.

2   Select the Favorites location you want to view.

To delete an existing Favorite location

1   On the Add/Delete Favorites dialog box, select the Favorite location to be deleted.

2   Click **Delete**. The dialog box closes and the selected Favorite is deleted from the list.

Note: The favorite address is displayed in the Address field and cannot be edited.

Determining which table columns to display

Tables are used to display information about a set of related entities in the Main and Association areas. It is possible to determine which columns to display in the Association area tables.

To determine which columns to display in the Association area

1   Click the Table icon on the upper right-hand side of a table and select **Column Chooser.**

2   In the Table columns dialog box, click the arrows to move the names of the columns that you want to display to the Visible box and the ones that you do not want to display to the Invisible box.

3   Click **OK**.

Copying data to the clipboard

At times you may want to save data displayed in the table area in a Microsoft Excel spreadsheet for further analysis or save an image of a graph to the clipboard.

To copy data displayed in the Association area to the clipboard

1   Click the Table icon on the upper right-hand side of a table and select **Copy to clipboard**.
      The table can be pasted into Microsoft Excel or as an HTML file.

To copy a graph to the clipboard

1   Right-click a graph and choose **Copy to clipboard**.
      You can now paste the image into any application that works with the clipboard.

Exporting to the Precise Custom Portal

The Export to the Precise Custom Portal Portlet feature enables you to export the view of the chosen table or graph and generate a portlet with that view in the Precise Custom Portal, so that it will provide you with another way of monitoring your application.

Prerequisites

To be able to use this feature, you need to have the following rights in Precise:

•   View permissions to all Tiers in the application

If you do not have sufficient rights, you will get an error message when trying to execute this feature.

Exporting the information

You can either export a table view or a graph view.

Note: The name field has the following restrictions: maximum 100 characters.

To export a table view

1   Click the Column Chooser icon.

2   Select **Export to the Precise Custom Portal Portlet**.

3   Insert a name in the name field that clearly describes the table view.

4   Click **OK**.

To export a graph view

1   Right-click the graph.

2   Select **Export to the Precise Custom Portal Portlet**.

3   Insert a name in the name field that clearly describes the graph view.

4   Click **OK**.

About role management in Precise for SQL Server

The Roles management feature in Precise for SQL Server lets you assign roles to users and control whether they will be able to do the following operations:

- **Whatif**. Lets you run WhatIf simulations on statements that Precise for SQL Server has collected, without having to physically make the change (depends on what is configured for Gen_Estimated_Plan and Recommend).

- **Recommend**. Lets you generate index recommendations (depends on Gen_Estimated_Plan).

- **Gen_Estimated_Plan (explain)**. Lets you generate a new execution plan.

- **Launch_MS_Tool**. Lets you launch external SQL tools (such as, Query Analyzer).

It is possible to grant permissions to all roles or to revoke permission from individual roles. Permission is a combination of permission type (such as: INSTALL, ADMIN, MONITOR) and operation (such as: VIEW, UPDATE).

Certain types of permission enforce permission dependencies. For example, if you grant a TUNE.RECOMMEND permission it will automatically grant TUNE.EXPLAIN. In the same manner, if you revoke TUNE.EXPLAIN then the TUNE.RECOMMEND permission is also revoked.

You can configure permissions and roles in AdminPoint.

For more information on configuring permissions and roles, see the *Precise Administration Guide*.

Opening your Precise product from the Precise StartPoint

Precise is a Web-based application. You can access the Precise user interface using the Internet Explorer browser, version 6.0 or later. The syntax of the Precise URL address is: `http://<server>:<port>`, where `<server>` refers to the Precise FocalPoint server and `<port>` refers to the port number used by the GUI Web server. By default, the port number is 20790 (20760 for J2EE). For example: `http://beanotown:20790`.

This URL provides secure access to the StartPoint using authorized roles. From here, you can launch all Precise products. It gives you a quick overview of the status of your applications and access to the AdminPoint, where you can perform various management tasks (see the *Precise Administration Guide* for details).

Note: You must have local administrator privileges on the server where the StartPoint is running.

To open your product using Precise StartPoint

1   Type the address of the StartPoint user interface into the Address bar of your browser and Click **Enter**. The Precise login page opens. The login page provides secure access to Precise and to your specific product.

2   Specify your authorized role name and password. By default, both role name and password are `admin`. For more information about role names, see the *Precise Administration Guide*.

3   Click **Login**. The StartPoint page opens. This is the Precise home page.

4   On the Product Selection bar, from the drop-down list, select the product you want to open.

Usage concepts

This section includes the following topics:

- How information is collected
- About application performance tuning
- About SQL statement tuning
- About object tuning
- About I/O tuning
- About instance statistics tuning

How information is collected

Precise for SQL Server collects and displays performance data that enables quick and accurate performance monitoring and analysis. To understand the information displayed in the various graphs, tables and views, it is important to understand how information is collected.

About duration values

Precise for SQL Server displays duration values for several entities.

The following table describes the duration value for each type of entity.

Table 3-1   Duration values for entities

Entity	Description
Current session	The time that has elapsed since the session started (login time). This is not the time the session was active executing statements. For example, if a session was connected to SQL Server for one hour without executing any statement, the duration will still be one hour.
Programs (such as, Logins, Machines, DB Users, Databases, time units)	The elapsed time of the sessions executing the program. **Note:** The time is not measured from the login time (as it is in the Current session entity) but from the moment the Collector started. The average duration counter displays the average time each program has been running.
Statement (batch)	The time that elapsed while the statement was executed. This is very similar to the duration of programs (see above); but since the statement is almost always active, this duration also indicates the time the statement was active. The average duration counter displays the average execution time of the statement.

If Interpoint is installed (for SAP, PeopleSoft, Siebel or COM+), the duration and average duration fields have no significant meaning and can be ignored.

About session states

SQL Server provides reporting on many different types of events. Precise for SQL Server groups events according to logical relationships to enable you to easily pinpoint heavy resource consumers.

For example, when the Collector samples the SQL Server, it assigns a state for each connection. An application may wait for many events in SQL Server. To simplify analysis, and help you to easily identify the source of resource consumption, Precise for SQL Server groups various wait types into wait groups or states and defines the status of sessions currently connected to SQL Server by their states. The state is calculated from the following sysprocess columns: waittype, waittime, and status, and is divided into two types of states: In MS-SQL and Non-In MS-SQL.

Precise for SQL Server also provides statistics about the waits encountered by threads that are in execution. This information is only available for SQL Server 2005 instances. Diagnosing these counters can provide important information on performance issues throughout the system.

About In MS-SQL states

The In MS-SQL session states signify that the session is performing activities in the SQL Server.

The following table describes In MS-SQL session states.

Table 3-2   In MS-SQL session states

Icon	In MS-SQL State	Description
CPU	Using CPU	The session is currently executing and not waiting for any other event. Using CPU can indicate a performance issue if the value exceeds 90% of the In MS-SQL value. When SQL Server does read ahead, the session does not suffer from I/O wait, but may still show a great deal of CPU usage.
I/O	I/O Wait	The session is waiting for an I/O operation to be completed or terminated. I/O operation is much slower than CPU operation so it is very important to monitor whether an application is suffering from I/O wait. I/O can be easily resolved by reorganizing files across disks or by adding new disks to your system.
TMP	Tempdb Wait	The session is waiting for an I/O operation or a buffer wait on tempdb pages. A high value may indicate a bottleneck in tempdb. 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. SQL Server 2005 uses tempdb for row versioning and online index builds, in addition to managing temporary objects, sorting and hashing.
	Lock Wait	The session is waiting for a lock to be released. The lock is held by another session.
	Redo Log Buffer Wait	The session is waiting for an operation of the log file to terminate. This state is generally encountered during a commit or rollback operation. A high log wait value may indicate a problem with the commit frequency in the processes experiencing massive updates to data.
	Memory Wait	The session is waiting for an internal resource to be freed. This state is divided into substates.
	Remote wait	The session is waiting for a remote query to terminate. Some of the DBCC commands that check the database pages also create this type of wait. A remote resource can be either SQL Server and Oracle.
CLR 10110	CLR Wait	The session is currently performing a common language runtime (CLR) execution and is waiting for another CLR task or event to be initiated or completed. All CLR-related waits are grouped together under this category. This state is only available for SQL Server 2005 instances.

Table 3-2   In MS-SQL session states

Icon	In MS-SQL State	Description
	Buffer Wait	The session is completing a network I/O operation.
		A high value can occur if results from a large result set are transferred to the client using shared memory netlibs or TCP/IP. In this case it is important to verify that only required rows and columns are returned.

About internal waits

The Internal Waits view displays internal waits, indicating that the session is waiting for an internal resource to be freed. An Internal Wait is divided into substates.

The following table describes the internal wait substates reported by Precise for SQL Server.

Table 3-3   Internal wait states

Internal State	Description
Buffer Pool	Groups together the events that show contention on pages in the buffer pool.
	Buffer wait on tempdb pages are considered to be part of tempdb wait.
Latch	The session is waiting for an internal lock to be released.
Parallel	The session is waiting for one of its sub-threads to complete its operation.
DTC	Aggregates waits that occur when Distributed Transaction Coordinator (DTC) sessions have to wait for each other.
	This state is only available for SQL Server 2005 instances.
DB Mirror	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.
	This state is only available for SQL Server 2005 instances.
Profiler	Aggregates a number of states associated with the SQL Profiler and lets you see how much of the database resources it consumes.
	This state is only available for SQL Server 2005 instances.
Memory	Aggregates several types of waits that indicate that a session is waiting for memory to be allocated to it.
	This state is only available for SQL Server 2005 instances.
Backup	Includes the wait type that commonly occurs when a Backup command is performed.
	This state is only available for SQL Server 2005 instances.
Other Internal waits	Aggregates the following types of waits: ■ Full text waits Includes wait types dedicated to the full text indexing service and appears whenever a full text index is in progress. ■ HTTP waits Includes waits that occur when HTTP and SOAP operations are executing. ■ Query notifications Aggregates a number of states associated with the synchronization of Query Notification sessions. This state is only available for SQL Server 2005 instances.

About lock waits

The Locking view displays a breakdown of lock waits, indicating that the session is waiting for an application lock to be freed. Lock waits are divided into substates.

Table 3-4   Lock wait states

Lock Wait	Description
Row lock	The lock is being held on a row in a heap table.
Key lock	The lock is being held on a key in an index or a row in a clustered index.
Page lock	The lock is being held on a page.
Table lock	The lock is being held on the entire table.
Other lock	Includes locks on database, extents and files.
MD statistics lock	Indicates a metadata lock is being held on the statistics of a table or index.
MD partition lock	Indicates a metadata lock is being held on a partition function or partition schema.
MD other lock	Indicates a general metadata lock is being held.

About Non-In MS-SQL states

The Non-In MS-SQL session states signify activities performed outside of the SQL Server.

The following table describes Non-In MS-SQL session states.

Table 3-5   Non-In MS-SQL session states

Icon	State	Description
	Request Wait	The session is waiting for the client to issue a request. A high value can indicate that a connection pool is too large and that many connections have been left open and are not active.
	Parallel Exchange Wait	The thread in a parallel session is waiting for data exchange from another thread.
	Idle	Indicates that the session is waiting for something other than a user request or is executing the WAITFOR DELAY command.

About Statement identifiers

Each statement is identified by the following parameters:

Text   The actual statement text.

Database   Each statement is associated with a single database.

User   Each statement is associated with a single user (also called the parsing user).

For example, the statement "select * from sysobjects" executed in the master and msdb databases is represented by two different statements in the product. The ability to roll up the performance of those statements to one is also important. This is supplied using other identifiers such as the program entity.

How Precise for SQL Server monitors availability

The Precise for SQL Server Collector monitors SQL server every second and collects data regarding resource consumption. If the instance status (Up/Down) was changed, a message code is sent by the Collector and causes the current availability of the instance saved in the PMDB to be changed. In addition, the availability of the instance is saved, over time, in the PMDB. During a planned downtime period the instance availability is showed as it was available during this period.

Note: If the Precise for SQL Server Collector is down, the instance availability status reported is "Unknown".

About application performance tuning

Precise for SQL Server allows you to proactively monitor, analyze, and tune SQL Server instances. By capturing, measuring, and correlating performance metrics from all critical system components. Precise for SQL Server provides a complete view of your application's performance. It detects business performance problems and helps you find suitable and comprehensive solutions thereby ensuring that your business applications always perform at peak efficiency.

Detailed information on the resource consumption of an application is extensive and includes data on CPU consumption, I/O waits, internal database waits, and so on. Still, with Precise for SQL Server you can easily identify the top resource consumers at different levels and at different times.

For example, you may want to start at the Instance level and then drill down to different Login names and the SQL statements they executed. You can view the status of current sessions and examine their consumption during the last minute, or focus your analysis on a particular time period, such as yesterday, last week, or last year. You can also examine changes in resource consumption patterns over time in different time units, such as:

- 15 minutes (time slice)
- hours
- days
- weeks
- months

About viewing current activity

The Current tab displays near real-time information on all sessions that are connected to your SQL Server instance during the last minute. Use the Current tab to solve bottlenecks, such as those caused by locked sessions or runaway processes. The Current tab displays activity information up to the last time slice of activity.

You can select an SQL Server instance and examine its resource consumption during the last time slice, such as CPU usage, I/O wait, and lock wait.

The application resource consumption is displayed on the instance level (as shown above) or as a summary of the session level of resource consumption (as shown below).

If you identified a blocking situation that is currently in progress (for example, there is a session waiting for a lock), select the Locks view to view the current sessions that are blocking the other sessions. The information is displayed in a tree format, which helps you easily identify which are the locked sessions, who they are being locked by and what is the lock level. You can drill down to the sessions involved in the lock to obtain more information about what is causing the lock. For example, you can drill down to a locking session to determine which SQL and which program are currently being executed.

About analyzing instance performance

While the Current tab displays near real-time data, it is recommended to begin the actual tuning process is the Activity tab.

The Activity tab allows you to analyze performance behavior over time or when the problem is reported (since in many cases the performance problem may no longer be evident from the data displayed in the Current tab). Using the Activity tab, you can analyze what happened at a specific point in time, identify problematic time periods, and drill down to analyze a bottleneck that caused a performance problem. You can begin by selecting an Instance and viewing the In MS-SQL overtime graph to identify problematic time periods.

In addition, long-term analysis is also required for understanding performance patterns and predicting future resource consumption. You can use the Activity tab to detect resource consumption patterns and suspicious deviations in past behavior.

The Activity tab can also be referred to when planning your system's capacity, performing period-to-period comparisons, and determining how best to optimize storage. Use the Activity tab to examine resource consumption of your SQL Server instance, over time, as well as the resource consumption of your databases, programs, logins, and so on.

You can also identify dominant resource consumption by selecting the Overview view.

The Overview view provides a breakdown of the In MS-SQL states, such as, CPU usage, I/O waits, Lock waits, and Remote waits. Examining this data allows you to pinpoint the type of resource causing the bottleneck. If for example you identify that your SQL Server is suffering from an I/O wait problem, upgrading your server to a much more powerful machine will probably fail to provide you with the expected results.

Note: A high CPU rate exceeding 90% may indicate that indexes are not being used.

About identifying resource consumers

Precise for SQL Server lets you drill down to additional levels to determine which entities are responsible for a particular performance problem. In the Association area you can observe different entities, such as databases, login names, users, programs, and client machines, to pinpoint the application component that consumes the most database resources.

For each entity (as, for example, the program entity), you can access the following information:

- Entities that are related to the entity (such as, the logins or machines that executed the program).

- Resources consumed by the entity. This enables you to determine if the entity is CPU or I/O bound, or waiting for a lock.

- Resource consumption patterns, which allows you to observe and identify trends in resource consumption. You can drill down even further to other entities associated with the selected entity in the Association area. For example, once you've identified a problem in a specific program you can drill down to the batches or statements the program issued, to determine the source of the problem.

About correlating with application server transactions

Precise for SQL Server lets you correlate between SQL Server resource consumption and internal application server data, as follows:

- For SAP applications you can drill down to the most resource-consuming SAP transaction, SAP user, SAP work type, and SAP application server name.

- For COM+ applications you can observe information on the most resource-consuming COM+ application, COM+ component, COM+ method, and COM+ interface.

- For PeopleSoft applications you can identify the most resource-consuming PeopleSoft Client Machines, PeopleSoft User IDs, Tuxedo OS Users, Tuxedo domains, and work types.

- For Siebel application you can identify the most resource-consuming Siebel Application, Views, Batch Components Groups, Batch Servers Components, Business Objects, Siebel users and work types.

These entities are extensions to the common application performance entities. You may use the Association area to drill down and investigate the relationship between various entities.

About associating with 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 or SQL statements that use few resources but are executed frequently.

For each statement or batch, you can access the following information in Precise for SQL Server:

- Entities that executed the statement or batch.

- Resources consumed by the statement or batch, which lets you determine if the statement or batch is CPU or I/O bound, or waiting for a lock, and so on.

- Resource consumption patterns that allow you to observe if the statement or batch execution is affected by excessive use of the same resource, by another statement or batch.

The drill down to most resource-consuming SQL statements or batches is also available from all COM+, SAP, PeopleSoft and Siebel extended entities.

About analyzing previous blocking situations

One of the main reasons that statements possess long execution times is that they are waiting for a resource that is locked by another session.

Using Precise for SQL Server, you can easily identify the following:

- Statements waiting for a locked object

- Locked objects (that is, contended objects)

- Sessions causing locks

In Precise for SQL Server, a statement waiting for the lock on a resource to be freed is in a Lock wait. The session holding the lock is called a Blocker session.

About SQL statement tuning

Most problems occur in SQL statements or batches. Precise for SQL Server lets you easily identify the most resource-consuming SQL statement or batch and view a detailed explanation of the execution plan of the SQL statement or batch. It can also help you resolve the problem by recommending which index would provide a better execution plan and reduce resource consumption.

If you are running SQL Server 2005 you can use the What-If option to check how creating a new index would influence the running of the application.

About identifying the most resource consuming statements or batches

Begin with the SmarTune tab to view a list of statements that require tuning. In the SmarTune tab, you can view expert knowledge regarding any area of your application showing performance deterioration and obtain comprehensive advice on how to solve specific problems. The Findings area lists the problematic statements while more comprehensive information is displayed in the Details area.

From the SmarTune tab you can launch, in-context, to specific displays in other tabs, to continue your analysis of the resource-consuming statement. The SmarTune tab provides guidelines that can help you narrow down your analysis and reach the root of a problem quicker and easier.

In the Association area of the Activity tab, you can associate to a specific database entity and then from there to its associated SQL statements.

When selecting an SQL statement, you can view a breakdown of its resource consumption over time, its text, and all locking contentions that it participated in (time spent waiting for a resource).

About collapsing constants into bind variables

When an application uses constant values and literals instead of bind variables (parameter markers), many different SQL statements actually constitute the same SQL statement with different values.

When each of these similar SQL statements is treated as a different statement, it is hard to summarize their resource consumption to identify the best candidate for SQL tuning.

Precise for SQL Server provides a unique algorithm that ignores constant values and groups all similar statements into collapsed statements. This enables you to identify the most resource-consuming collapsed statement.

The performance data for the collapsed statement is a summary of all performance data for all statements in the group. Most of the time, all the statements that were grouped together into one collapsed statement have the same execution plan. However, different values in the WHERE clause may lead the optimizer to choose a different execution plan. Precise for SQL Server lets you select a Collapsed statement and associate to all the different execution plans of the SQL statements that were grouped together.

About viewing an execution plan

To view the execution plan of the statement and understand how the optimizer chooses to perform the statement, launch to the SQL tab. Each branch in the execution plan represents one operation. Each operation is displayed using the following format:
[<execution order>]<operation type>(<options>)<accessed object>
[<cost in percentage>]
When examining the execution plan, always start by exploring it from right to left and top to bottom (according to the execution order in a branch on the tree). The first operation to be performed has execution order 1, followed by 2, and so on.

About estimating the cost of an execution plan

By observing the access plan of a selected statement or batch, you can analyze the access path chosen by the SQL Server Optimizer.

From the Instance Details area, in the Dashboard tab, you can observe information reported for the top statements found by the SmarTune process. You can launch directly from a finding in the Findings table to the SQL tab to compare the costs of the execution plan operators and quickly locate the most resource-consuming operation in the execution plan.

Operations are evaluated as follows:
- Statements with total estimated costs greater than the predefined thresholds (where 1 is the default) are checked for critical operations.

- Operations whose estimated cost percentage is greater than a predefined threshold (where 20% is the default) are marked as critical.

The information displayed in the Operations tab helps you understand the nature of the execution plan, its major operations, and the major access types to objects (such as, index scan, index seek, and table scan).

About analyzing changes and their effect on the statement's execution plan

Any change made to the schema or configuration parameters can adversely affect the performance of your application. Precise for SQL Server allows you track the schema and configuration changes and compare them with the resource consumption of your statement to understand how the changes related to the statement affected its performance and execution plan.

Check the Findings table in the SmarTune tab to view information on schema changes. Select a finding and launch to the SQL History view to observe the effect these changes had on resource consumption.

As you can see schema changes, volume changes, and database option changes, were made to objects referred by this statement. By examining the time the changes were reported and when the statement's performance began to deteriorate, you can locate the changes that may have been the cause of the degradation in statement performance. In the Association area you can view additional details regarding the changes and you can launch from a specific change to another tab to further your investigation. (For example: by clicking on the "Launch icon" icon of "Instance parameter changes" you will launch to Object tab while displaying the Instance Parameters Changes" in the Main Area.

About recommending indexes for an SQL statement

Precise for SQL Server uses the Microsoft® SQL Server 2000 Index Tuning Wizard (ITWIZ) and the Microsoft® SQL Server 2005 Database Tuning Adviser (DTA) to obtain recommended indexes and statistics for a selected statement or batch. The Recommended Indexes/Statistics list is displayed in the list on the left. Detailed information for each recommended item, such as a list of key columns and the DDL "create index/statistic" text, is displayed in the information tabs on the right.

If you are running SQL Server 2005, use the What-If option to show the statements whose performance would improve and the statements whose performance would deteriorate by implementing the recommendation.

About object tuning

In many cases, you may want to tune a database object rather than a specific statement that accesses the database object. This is especially true if you were not the one to write the application (for example, ERP and CRM applications) and therefore cannot change the text of any statement. In this case you will want to identify which objects are the best tuning candidates.

About identifying the best candidate for object tuning

In the Explore Objects tab in the Objects tab, you can examine the performance of active objects and determine which objects are good candidates for tuning. You can observe whether an object is an index or table, the name of the database it is associated with, a summary of the time it spent in SQL Server, and its size (in pages). If the object is an SQL Server 2005 object, additional information regarding I/O operations and waits, contentions, index usability, and logical, user and system operations is also displayed.

Clicking on the launch icon, next to an object, launches to the Recommend tab where you can view recommendations on the selected table.

You can also drill down on an object and view additional information on the selected object in the Main area of the Explore Object tab. For example, you can drill down to associated SQL statements. This helps you identify the top resource-consuming SQL statements that are accessing a selected table or index.

You can view all SQL statements that use the indexes built on the specific table. Drilling down from an index to all SQL statements using the selected index allows you to determine whether or not the specific index is actually being used. Precise for SQL Server also displays a list of unused indexes for a table or database.

Another method you can use to investigate how an index is being used is to drill down on the index and view information on its common access patterns. If you observe that an index is being updated, but shows no seek or scan activity, this may indicate that the index is not being optimally used.

The following indicators may indicate that you need to re-evaluate your index:

- User operations are low while system operations are high, indicating that the index is only being used for maintenance purposes.

- Level of I/O contentions and the amount of statement access time is high.

About recommending an index for a table

The decision to add a specific index to a table is complex. The index can improve one statement and may negatively affect others. Therefore, before adding an index, you should examine every statement accessing the table to see how the new index will impact it.

Precise for SQL Server uses the Microsoft® SQL Server 2000 Index Tuning Wizard (ITWIZ) and the Microsoft® SQL Server 2005 Database Tuning Adviser (DTA) to obtain recommended indexes and statistics for a selected table. The recommendations are based on all the statements referring to the selected table that were executed during the selected time period that have an average duration time greater than a predefined threshold (0 seconds is the default).

The Recommended Indexes or Statistics list is displayed on the left. Detailed information for each recommended item, such as a list of key columns and the DDL "create index/statistic" text, is displayed in the information tabs on the right. In addition, if you are running SQL Server 2005, Precise for SQL Server can recommend whether to drop an index, or create a partition function or schema.

If you are running SQL Server 2005, use the What-If option to show the statements whose cost would increase and the statements whose cost would decrease if the recommendation was implemented. You can select the recommendations you want to test and run the What-If function specifically on those recommendations.

The What-If function displays a pie chart that indicates the distribution of the cost of all statements that were evaluated. The Analyzed Statements table displays the explained statements that access the objects whose recommendations are being evaluated. You can continue your analysis of a particular statement on the SQL tab Plan tab, by clicking its statement ID link.

About examining schema changes over time

The Collect Schema Changes process collects changes made in an instance configuration, databases, objects, columns, indexes, and index keys. The objects sampled are User Tables, Stored Procedures, Views, Functions (all kinds), Foreign Keys, Triggers, and Extended Stored Procedures.

Schema changes take the following aspects into consideration:

- Entities dropped

- Changes made to the existing entities (collected properties are detailed in the Objects tab)

- Entities created

The changes are displayed in a list (summarized and detailed) and an overtime graph. The changes are summarized by groups such as Schema Changes, Instance Property Changes and Database Option Changes at the Instance level and Object Change in Structure and Index at the Database level. The deeper you drill down into the sub-entities the more detailed the information you will see regarding the changes. For example, you can drill down to a specific database to observe database option changes.

In addition, the In MS-SQL overtime graph is displayed with respect to changes so that you can observe how the changes that were made affected the performance of your application.

About examining object space usage over time

Another way to tune a database object is to examine its space usage. The Collect Space Utilization process collects table and index space information daily (by default).

Space usage information allows you to examine space allocated versus space used by a table or an index. This lets you easily keep track of object growth over time and decide how to allocate memory and connect objects to file groups in the specified database.

About I/O tuning

In theory, database files should be spread over different physical disks so that the I/O for one database file will not interfere with the I/O of a different database file. As a result, tuning the I/O on a large storage machine is a complex task. Moving a database file from one disk to another can speed up some parts of the application but may slow down others.

Precise for SQL Server includes storage points that can help you tune the I/O in your system. Storage points are agents that sample activity in the Storage units against which they are installed. They map datafiles to Storage devices and provide data that relates I/O wait of database objects to the Storage devices on which they are stored.

If you are running SQL Server 2005, both I/O read and I/O write information is displayed in the Objects tab.

Note: Storage points are extensions to Precise for SQL Server. They require an additional license.

About identifying overactive storage devices

Precise for SQL Server displays information that relates I/O wait to the EMC Storage devices in which they are stored. This allows you to locate specific EMC Storage devices where performance problems are occurring.

By analyzing the information collected from your SQL Server databases, Precise for SQL Server provides an accurate picture of I/O activity and all the information needed to tune the I/O system. By identifying which EMC Storage devices are overactive, you can maintain performance levels by reorganizing storage configurations, reallocating your database objects to other file groups, or moving file groups to alternative devices.

About identifying and balancing I/O waits on the storage device level

After identifying which storage devices suffer from performance bottlenecks, you can drill down to them and analyze the I/O waits experienced during the selected time period. This can help you identify the logical files that are most probably causing the I/O bottleneck.

About identifying the most accessed database files

When one of the storage points is installed, you can compare the I/O wait on different physical disks to find out where the bottleneck is. You can then drill down from the physical disk level to the database file level to identify the database file with the most I/O waits.

About identifying and balancing I/O waits on the physical device level

Before moving a database file to another physical disk, you can examine the I/O wait on the database file to identify the time period that is experiencing the most I/O waits. You may also want to examine I/O waits on a physical disk over time to identify which physical disk is experiencing either no I/O waits, or very few I/O waits, during the same time period.

About examining file space usage over time

The Collect Space Utilization process collects database file space information daily (by default). This lets you examine file size versus the space used by each file in a database. This can help you locate the most space-consuming file, check its rate of growth, and help you determine how to best allocate memory and arrange the database files on disks.

About instance statistics tuning

The Precise for SQL Server Collector collects a wide range of performance counters from the operating system, some which belong to SQL Server and some to the operating system (OS) itself. These performance counters are sampled every minute and displayed in the Current and Statistics tabs. When loaded into the PMDB, the counters are aggregated and kept at a slice level. The counters are then aggregated to daily, weekly and monthly levels.

About observing instance statistics over time

The Statistics tab provides many performance counters grouped into several predefined overtime graphs that enable you to locate various aspects of performance problems, such as CPU, paging, I/O, and network. The Statistics tab can be used to monitor your system's current state as well as previous states. It is also possible to monitor a specific counter associated with a particular performance group by selecting the required counter.

Getting an overview of your application

This section includes the following topics:

- About the Dashboard tab
- How the Dashboard tab is structured
- How the Dashboard can help you identify performance problems

About the Dashboard tab

The Dashboard tab provides an overview (approx. 70-80%) of the information displayed in other Precise for SQL Server tabs. The Dashboard is the starting point for your Precise analysis since it displays an overview of all the instances available in the selected application and information that is also available in some of the other tabs.

The Dashboard tab displays availability, performance findings, statistics and resource consumption of all SQL Server instances for the selected time period. The Current view in this tab can be used to monitor an instance's current state, display a list of its active sessions, and its resource consumption during the last minute. In addition you can view statistics, resource consumption and top resource consumers, for the selected time frame.

The Dashboard tab also displays information from the SmarTune tab. SmarTune can provide you with expert knowledge regarding a specific area of deterioration, and comprehensive advice on how to solve a specific problem and optimize performance. This is an important tab to begin with if you want to start performance tuning on a database.

The Dashboard tab provides answers to the following types of questions:

- Are all my SQL Server instances up?
- Which instance is consuming the most resources?
- Which program is consuming the most resources for the specified instance and across all instances?
- From what types of performance problems is an instance suffering?

The All row in the All SQL Server Instances table located in the Main area (left pane) displays the average availability of all instances. When you select the All row a summary of the activities of all the instances and a comparison of application identifiers is displayed in the various tab pages viewed in the Instance Details area (right pane).

In general, launching to a tab from the All row focuses in on the Tier entity, which summarizes the data collected for all instances. Tabs that do not have a Tier entity open on the default instance.

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

See the following topics for more information:

- About Precise for SQL Server tabs
- About the Current tab
- About the Activity tab

- About the Objects tab

- About the Statistics tab

- About the SQL tab

- About the Maintenance tab

- About the SmarTune tab

How the Dashboard tab is structured

The Dashboard tab is divided into two areas—the Main area and the Instance Details area. The Main area lists all the instances that are monitored by the Precise for SQL Server Tier. The Instance Details area provides comparative information regarding the selected instance.

The information displayed in the Dashboard tab is influenced by the time frame you selected. See Selecting a time frame.

About the Main area in the Dashboard tab

The instance table displayed in the Main area lists all the instances that are monitored by the Precise for SQL Server Tier. Each row corresponds to an instance. The All row displays the average availability (%) of all instances. The following table shows the information displayed in the Instance table.

Table 4-1   Instance table

Column	Description
Icon	Indicates whether or not the instance is available.
Instance	Name of the instance.
In MS-SQL	Breakdown of In MS-SQL duration of the instance.
Executions	Total statements executions.
Availability	Average percentage of time the instance was available.

About the Instance Details area in the Dashboard tab

Clicking on the various tabs in this area allows you to display overall activity, top performance findings, active sessions, current and statistical information for the selected instance. Clicking on the hyperlink for specific data items launches, in context, to the related tab from which the data was derived.

See Configuring Findings settings.

The following table describes the information displayed in the Instance Details area.

Table 4-2   Information displayed in the Instance Details area

Tab	Information displayed
Overview	The Overview tab displays In MS-SQL data and performance findings for the selected instance.
Applications	The Applications tab displays the top resource-consuming databases, programs and machines.

Table 4-2   Information displayed in the Instance Details area

Tab	Information displayed
More	The following options are available in the More tab: ■ The Statistics tab displays statistical information for the selected instance, as reported by the SQL Server and collected by the Precise for SQL Server Collector. ■ The Activities tab displays the top resource-consuming statements, logical files and storage devices. ■ The Availability tab displays the average percentage of time the instance was available, and a detailed log of the changes in availability status. ■ The Current tab displays the overall activity and active sessions that occurred within the last minute for the selected instance. **Note:** The Current tab is not available if All Instances is selected.

Note: If you select the All row, the information displayed reflects the average availability of all instances. The "Current" option and the Findings table will not be available.

About gaining an overview of your system's performance

The Overview tab displays In MS-SQL data and performance findings for the selected instance. You can continue investigating a problem as follows:

- Click In MS-SQL to launch the Activity tab with an instance or Precise for SQL Server Tier context.
- Click Findings or a launch icon in the Findings table to launch to the SmarTune tab with the selected finding in context.

The following table describes the information displayed in the Overview tab.

Table 4-3   General instance data

General instance data	Description
In MS-SQL	Displays a breakdown of the total In MS-SQL duration over time, into states such as, Using CPU, I/O wait, and Lock Wait.
Findings	Lists all the statements, collapsed statements, tables or instance events that were detected for a specific instance. All problems are ranked by severity. The information displayed in the findings table is also displayed in the SmarTune tab.
Details for ... Finding	Provides quick links to additional tabs and views that would be helpful in your analysis of a particular problem.

About observing top resource-consuming entities

The Applications tab displays the top resource-consuming databases, programs and machines.

All entities in this display launch the Activity tab, in context of the instance.

The following table describes the information displayed in the Databases table.

Table 4-4   Database applications

Column	Description
Database	The name of the top resource-consuming databases.
Sessions	Total number of sessions opened.
In MS-SQL	The resource consumption of the database broken down into states such as, Using CPU, I/O wait, Lock wait states. Click on the respective icons to display the data in this column in a stacked graph, bar graph or numerical format.

The following table describes the information displayed in the Programs table.

Table 4-5 Program applications

Column	Description
Program	The name of the top resource-consuming programs.
Sessions	Total number of sessions opened.
In MS-SQL	The resource consumption of the program broken down into states such as, Using CPU, I/O wait, Lock wait states.
	Click on the respective icons to display the data in this column in a stacked graph, bar graph or numerical format.

The following table describes the information displayed in the Machines table.

Table 4-6 Machine applications

Column	Description
Machine	The name of the top resource-consuming machines.
Sessions	Total number of sessions opened.
In MS-SQL	The resource consumption of the machine broken down into states such as, Using CPU, I/O wait, Lock wait states.
	Click on the respective icons to display the data in this column in a stacked graph, bar graph or numerical format.

About examining statistics for an instance

The Statistics tab, accessed from the More tab, displays statistical information for the selected instance, as reported by the SQL Server and collected by the Precise for SQL Server Collector.

All items in this display launch the Statistics tab with an Instance or Precise for SQL Server Tier context.

The following table describes the information displayed in the Statistics tab.

Table 4-7 Instance statistical data

Instance statistical data	Description
Server CPU	Displays the SQL Server instance CPU utilization for the selected instance with respect to the CPU utilization of other processes (non-SQL Server).
Load	Displays instance load, for the selected instance, broken down into batch requests/sec, transactions/sec, and logins/sec.
Page faults	Displays memory paging of the selected instance vs. memory paging caused by other processes (non-SQL Server).

About viewing top resource-consuming statements, logical files and storage devices

The Activities tab, accessed from the More tab, displays the top resource-consuming statements, logical files and storage devices.

All entities in this display launch the Activity tab, in context of an instance or Tier.

The following table describes the information displayed in the Statements table.

Table 4-8 Statement activities

Column	Description
Statement	Displays the ID of the top resource-consuming statements.
Duration (Avg)	Displays the average duration of the statement.
In MS-SQL	Displays the resource consumption of the statement broken down into states such as, Using CPU, I/O wait, and Lock wait states. Click on the respective icons to display the data in this column in a stacked graph, bar graph or numerical format.

The following table describes the information displayed in the Logical Files table.

Table 4-9 Logical files activities

Column	Description
Database	Displays the name of the file's database.
Logical File	Displays the logical name of the file.
I/O Wait	Displays the total I/O waits registered for the file. Click on the respective icons to display the data in this column in a stacked graph, bar graph or numerical format.

The following table describes the information displayed in the Storage Devices table.

Note: This information is only displayed if storage points are installed.

Table 4-10 Storage device activities

Column	Description
Device Type	Indicates a type of storage device, such as, Windows, HP, Symmetrix.
Unit ID	Displays the ID of the device's unit.
Device	Displays the name of the device.
I/O Wait	Displays the total I/O waits registered for the device. Click on the respective icons to display the data in this column in a stacked graph, bar graph or numerical format.

About viewing the availability of SQL Server instances

The Availability tab, accessed from the More tab, displays the average percentage of time the SQL Server instance was available (that is, it met the SLA Availability requirements configured in the SLA settings over the specified time period) and a detailed log of the changes in availability status.

All entities in this display launch the Activity tab, in context of the instance or Tier.

The following table describes the information displayed on instance availability.

Table 4-11   Instance availability

Item	Description
Availability	The average percentage of time the SQL Server instance was available.
Log	A detailed log of the changes in availability status.

About viewing your system's current activity

The Current tab displays the overall activity and active sessions that occurred within the last minute for the selected instance.

All items in this display launch the Current tab, in context of an instance.

The following table describes the information displayed in the Session Activities area.

Table 4-12   Sessions

Item	Description
Sessions	Total number of sessions connected to the selected instance.
Active sessions	Total number of sessions currently active.
Locked sessions	Total number of locked sessions.

The following table displays information on the active sessions that occurred within the last minute, for the selected instance.

Table 4-13   Active sessions (last minute)

Column	Description
Icon	Represents the current session's state (such as, Lock Wait, Request Wait).
SPID	The session ID.
Program	The name of the program that opened the session.
State	The current state of the session.
In MS-SQL	Total In MS-SQL entities for the session, occurring within the last minute, broken down into states (such as, Using CPU, I/O wait, and Lock Wait).

The following table describes the information displayed in the Current Activity graphs.

Table 4-14   Current activity graphs

Item	Description
Overall activity (Last minute)	Provides a graphical display of the overall activity that occurred within the last minute, for the selected instance, broken down into the following states: Idle Wait, Parallel Exchange Wait, Request Wait and In MS-SQL.
In MS-SQL (Last minute)	Provides a graphical display of In MS-SQL activity that occurred within the last minute, for the selected instance, broken down into states, such as, Using CPU, I/O wait, and Lock Wait.

About the Findings area in the Dashboard tab

The Findings area in the Dashboard tab displays performance problems detected in the select instance(s). The findings list is displayed according to severity (highest to lowest).

To view the details of a specific finding, hover over the finding name. The area for the selected finding will expand, displaying information and recommended actions. Click on the relevant link to launch the appropriate tab in context, or **Learn more** to open the help for that finding.

For more information about how you can identify specific performance problems, see How the Dashboard can help you identify performance problems.

How the Dashboard can help you identify performance problems

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

- Identifying the most resource-consuming instance
- Analyzing top performance findings per instance
- Identifying the highest resource consumers in various entities
- Examining instance statistics
- Monitoring a session's activity during the last minute
- Checking SQL server instance availability

Identifying the most resource-consuming instance

Precise for SQL Server enables you to determine the most resource-consuming instances, indicating a potential performance problem.

To identify the most resource-consuming instance

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

2   In the All SQL Server Instances table, observe the total In MS-SQL statistics and statement executions of the instance you want to analyze.

3   Select the row of the instance you want to analyze.

4   In the Instance Details area, on the Overview tab, check the In MS-SQL graph to view its resource consumption, over time, to identify problematic time periods and observe its general resource consumption trend.

5   Check the Findings table to view the top problematic findings per instance.

Analyzing top performance findings per instance

The Dashboard tab helps you pinpoint performance problems by letting you observe the top problematic findings per instance, statement or object. You can then drill down on a finding and continue your analysis in the SmarTune tab, which provides important guidelines and clues as to what may be causing the performance problem.

To analyze top performance findings per instance

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

2   In the All SQL Server Instances table, observe the total In MS-SQL statistics and statement executions of the instance you want to analyze.

3   Click on an instance in the All SQL Server Instances table.

4   In the Instance Details area, on the overview tab in the Findings table, observe the top findings for the selected instance. The findings displayed in this table are sorted by severity. Additional information and hot links for the selected findings is displayed in the Details area below the Findings table.

5   In the Type column, click on the link to the instance that you want to analyze further or on the **Findings** link located above the table.

6   In the SmarTune tab observe the data displayed on the finding in the various tables and graphs, and follow the links provided to view additional information on the problem in other tabs. The SmarTune tab also provides details and recommendations on how to solve a problem.

Identifying the highest resource consumers in various entities

Precise for SQL Server enables you to determine the most resource-consuming application components (such as, Databases, Programs, and Statements).

To identify the highest resource consumers in various entities

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

2   In the All SQL Server Instances table, determine which instance consumes the most resources.

3   Select the row of the instance you want to analyze.

4   In the Instance Details area, on the Applications tab, identify the highest application resource consumers for the selected instance.

5   In the Instance Details area, on the Activities tab, identify the highest activity resource consumers for the selected instance.

6   Launch to the Activity tab to continue the performance tuning process on the selected component.

Note: The entities displayed in the Application tab will differ according to the ERP installed and will take into account that ERP's specific entities. For example, if you have SAP installed, the following types of entities are displayed: Work Types, Logins, SAP Users and Programs, and SAP Transactions.

Examining instance statistics

You can examine instance statistics over time to confirm that CPU and memory usage are within normal ranges. A continually high CPU usage may indicate the need to improve the application or to add more processes to the machine to enhance performance. In addition, a high number of page faults may indicate an inefficient SQL statement or lack of memory.

To examine instance statistics

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

2   In the Instance Details area, click **More...** and select Statistics. Observe CPU usage vs. load (number of transactions, batches and logins) and paging statistics.

3   Launch to the Statistics tab to observe additional instance statistics that can help you locate performance problems in various other aspects of your system, such as I/O and Network statistics.

Monitoring a session's activity during the last minute

Monitoring a session's activity during the last minute in SQL Server, can help you locate activities that deviate from the norm, such as locking and latch waits.

To monitor a session's activity during the last minute

1   In the Instance Details area, click **More...**, and select Current, observe all active sessions (sessions that executed statements during the last minute) and the In MS-SQL breakdown into states, such as, Using CPU, I/O wait, and Lock wait states.

2   Launch to the Current tab to view additional information on current sessions. For example, if you identify a session suffering from lock wait during the last minute of activity, launch to the Current tab and analyze the locking tree to discover what is locking the session and why.

Checking SQL server instance availability

It is possible to examine the availability of SQL Server instances and observe their current status as well as their level of availability over time. In addition, you can determine whether their level of availability meets the SLA standard defined in the SLA availability setting in AdminPoint.

For more information, see the *Precise Administration Guide.*

To check SQL Server instance availability

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

2   In the All SQL Server Instances table, check the current availability state of the instance you want to analyze. A status icon, located in the first column, shows the current availability of the instance. Move the pointer over the icon to display a ToolTip showing the current status of the instance and the date its status changed.

3   Select the row of the instance you want to analyze. The average availability of the instance is displayed in the Availability column.

4   In the Instance Details area, click **More...** and select Availability. Check the Availability graph to view the availability of the selected instance with respect to the configured SLA and the log.

See How Precise for SQL Server monitors availability.

Getting current application information

This section includes the following topics:

About the Current tab

The Current tab displays near real-time information on all sessions that were connected to your SQL Server database during the last minute. If you have discerned an immediate performance issue and are looking for an indication as to where to begin your analysis, start with the Current tab. It is also worthwhile to start with the Current tab if your system appears to be 'hung' and there is no throughput.

You can use this tab to monitor your system's current activities and resource consumption and to identify performance problems, such as current locking problems and sessions involved in locks, as they occur.

The Current tab provides answers to the following types of questions: "How many sessions are currently running?" or "Why doesn't my application work when I click OK?" The Current tab also shows you which sessions are currently active, and which machines are connected to which users.

Using the Current tab, you can monitor current connections to your SQL Server instance, check the state of each connection, and identify the resources currently utilized by each session. Further investigation of each session provides a thorough understanding of the SQL statement being executed, the login name executing the statement, the program, and the resource utilization for the session. In addition, you can view the statements and batches executed by the selected session within the last time slice.

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

In the Current tab, data on a current session is continuously updated to provide information on what happened during the last minute or time slice. Because the displayed session information is derived from a dynamic application, it is recommended to regularly refresh the screen, either manually or automatically, to ensure that the information displayed is up-to-date.

Note: The Current tab displays information on the last minute of activity. For example, if you've worked at your screen for ten minutes and then refresh the information displayed on your screen, you will be able to view new information that occurred during the last minute of activity.

See Analyzing locks currently held.

How the Current tab is structured

The Current tab displays information on a selected entity and its associated entities. For example, if the parent entity is a Session, its associated entities would be Statements or Batches.

When you open the Current tab from the Dashboard tab, the selected entity is by default Instance, meaning that information is displayed on the instance level. When you open the Current tab from the Activity tab, the tab is launched with the selected instance or entity (in the case of an entity, this is only true of the database entity), meaning that the information is displayed on the instance or entity level. If you open the Current tab from another tab, the historical settings (meaning those settings which were selected when you left the tab, such as a drill-down to a Locked Objects entity) are taken into account and the information displayed the last time you viewed this tab is displayed (similar to clicking the History button and returning to a previous tab).

The selected entity is always reflected in the Tab heading, which serves as a point of orientation. The highest-level entity you can view information for in the Current tab is Instance. You can view another instance by selecting it from the Instance list.

About the Main Area in the Current 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, if you are analyzing an Instance entity, you can, focus exclusively on the data displayed in the overview, such as, In MS-SQL data (default), or choose to view performance statistics on the temporary database (Tempdb) in the SQL Server.

About the Association area in the Current tab

The Association area provides corresponding information on the entities associated with the selected entity (displayed in the Main area). For example, you can associate to the Locked Objects related to the selected instance. The selection you make is reflected in the Association area only; the Main area remains unchanged. For some entities, tabs above the Association area enable you to view additional information. For example, the tabs displayed for a database entity are: General, In MS-SQL and Statistics. Clicking a tab displays different table columns with the same type of child entities.

From the Association area, you can also drill down to another entity by clicking a table row. A drilldown affects the entire tab. When you drill down to another entity, the Tab heading displays the new selection, the Main area displays information on the newly selected entity, and the Association area displays the entities associated with the selected entity.

For example, when you want to view information on a specific session's holding locks, in the Current tab, choose Sessions Holding Locks from the Association controls. The Association area changes to display sessions holding locks-related information. Note that the Tab heading and the Main area remain unchanged.

If you want to view detailed information on a specific session holding lock, click the row of the sessions holding locks you want to view detailed information for. The Tab heading indicates the newly selected entity; the Main area displays locking and In MS-SQL information on the session you drilled down to, and the Association area shows information on the statements associated with this session.
See [About Precise for SQL Server tabs](#).

About examining entities in the Current tab

The Current tab displays information on different entities. This section provides an overview of all entities, their meaning, and their views.

The following entities can be examined in the Current tab:

- Instance

- Database

- Session

- Locked Object

- Session Holding Locks

- Statement and Batch

- Active Session

About the Instance entity

The Instance entity displays information on the resource consumption of the entire instance and allows you to focus on how it is performing, during the last minute.

About getting an overview of current Instance activities

The Instance Overview displays information on the activity of the instance, such as number of sessions and the resource consumption breakdown of the instance.

The following table describes the information displayed in the Instance Overview.

Table 5-1   Instance Overview view

View area	Description
Sessions	■ Sessions—Indicates the number of sessions (different SPIDs) that were connected to the SQL Server instance during the last sample. ■ Active Sessions—Number of sessions (different SPIDs) connected to the SQL Server instance during the last sample that were not in a state that is not part of the In MS-SQL group. ■ Locked Sessions—Number of blocked sessions in the last sample. ■ Memory Usage (Max)—Displays the maximum memory (in MB) consumed by all sessions that were opened in SQL Server during the last 15 minutes. ■ Open Transactions (Max)—Maximum number of transactions that were opened in SQL Server during the last 15 minutes.
In MS-SQL (Last minute)	Displays the resource consumption breakdown of the instance during the last minute (that is, 60 samples of the Collector). In addition to the resource distribution, the view displays the number of sessions in each state during the last sample. It is important to remember that since this view only displays the last minute's activities, you will always be viewing the activities that occurred during the last minute and not the activities that occurred since the last refresh action. The following information is displayed: ■ State—Displays the In MS-SQL state. ■ %—Percentage of time the selected entity was in this state. ■ In MS-SQL—Graphical representation of the Time column. ■ Time—Amount of time the selected entity was in this state. ■ Sessions (Last Sample)—Current number of sessions in the specified state. For example, there are currently three sessions experiencing a Lock Wait state. **Note:** All the information displayed in this table represents information collected during the last minute, with the exception of the Sessions column, which represents the number of sessions, in the last sample, that were in a particular state.

See About session states.

About viewing Instance Tempdb usage

Displays information on the current usage of the tempdb database.

The following table describes the information displayed in the Tempdb usage view.

Table 5-2 Instance Tembdb usage view

View area	Description
Temporary tables	Displays the following information on temporary tables: ■ Active—number of temporary tables that are currently used by any session. ■ Idle—number of temporary tables that are currently not used by any session. ■ Local—number of local temporary tables currently defined in the database. ■ Global—number of global temporary tables currently defined in the database.
Tempdb usage (Last second)	Displays statistical information on the tempdb database files within the last second. If the tempdb database contains more than one file, the displayed value is a total of all tempdb database files. ■ Reads—total number of reads that were executed in the tempdb database during the last second before the screen was refreshed. ■ Writes—total number of writes that were executed in tempdb database during the last second before the screen was refreshed. ■ Bytes Read—total number of bytes that were read during the last second before the screen was refreshed. ■ Bytes Written—total number of bytes that were written in tempdb database during the last second before the screen was refreshed. ■ I/O Wait (MS)—total amount of time (in milliseconds) that sessions waited for I/Os to complete in tempdb database.
Tempdb size	Provides a graphical representation of the components of the tempdb database size, broken down as follows: ■ Active—total size of all the temporary tables currently in use. ■ Idle—total size of all the temporary tables that are currently not in use. ■ Free Space—total free space available in the tempdb database. ■ Other Size—space currently used by other tables/operations. For example, system tables defined on the tempdb, space used by a sort operation currently being executed, or raw versioning.

About viewing Instance locks

Displays lock wait duration broken down into lock levels such as Table, Page and Key, for the selected Instance, during the last minute. The sessions that are currently involved in a locking or blocking situation are displayed in tree format.

Note: Since the Lock tree represents the locks that are currently occurring, and the lock breakdown represents the duration of locks occurring during the last minute, it is possible that a situation may occur whereby a lock was freed, but since the last minute has not yet passed, the lock tree will display the message 'No locks found' while the Lock Breakdown grid still shows lock duration.

The following table presents a detailed explanation of the Lock Tree and Lock Breakdown (Last Minute) mechanism.

Table 5-3   Instance Locks view

View area	Description
Lock tree	It is important to determine the following information for each blocking chain: ■ Which sessions are involved in the lock? ■ Which session is blocking the others? ■ Which object is blocked? ■ What type of lock is causing the block? The Lock Tree provides the necessary information to get to the root of the problem by providing information on sessions currently involved in the blocking situation. A yellow lock indicates that the session is currently locking 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 itself. To gain a better understanding of the lock tree take a closer 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 Lock Tree displays the SPID, program name, the object being blocked and the lock type.
Lock breakdown (Last minute)	Displays the Lock Wait duration broken down into lock levels such as Table, Page and Key, for the selected instance, during the last minute: ■ Lock Level—such as, Row, Key, Page, Table, or Other. "Other" includes all other locks. ■ Lock Wait—a graphical representation of the data displayed in the Time column. ■ Time—amount of time the selected instance was locked at the specified lock level. ■ %—percentage of time the selected instance was locked at the specified lock level. ■ Sessions (Last Sample)—current number of sessions locked at the specified lock level. **Note:** The table only displays lock levels that the selected instance was blocked by. If the selected instance was not blocked, the information area remains empty.

See <u>About session states</u>.

About viewing Instance statistics

Displays instance statistics for the last 30 minutes, such as CPU usage, memory paging and disk usage.

The following table describes the information displayed in the Instance statistics view.

Table 5-4   Instance Statistics view

View area	Description
Sessions	Displays the number of sessions connected to the instance, broken down into active and locked sessions, as follows: ■ Sessions—number of sessions (different SPIDs) connected to the instance during the last sample. ■ Active sessions—number of sessions (different SPIDs) that were in one of the In MS-SQL states during the last sample. ■ Locked sessions—number of blocked session in the last sample.
CPU usage	Displays statistics that report on CPU usage, broken down into SQL Server CPU usage and Non-SQL Server CPU usage.

Table 5-4   Instance Statistics view

View area	Description
Page faults	Displays statistics that report on page faults broken down into SQL Server Page Faults/sec and Non-SQL Server Page Faults/sec.
Load	Displays statistics that report on the instance load such as Batch Requests/sec, Transactions/sec and Logins/sec.
Disk operations	Displays statistics that report on I/O load, such as percentage of elapsed time the disks are busy servicing read/write requests.
Network packets	Displays statistics that report on network load, such as Max Packets Received/sec and Max Packets Sent/sec.

About the Database entity

Displays information on the activity and resource consumption of the database, including the number of sessions connected to the database during the last sample taken, various session-related statistics and the database resource consumption breakdown for the last minute.

About getting an overview of current Database activities

Displays information on the activity and resource consumption of the database, including the number of sessions connected to the database during the last sample taken, various session-related statistics and the database resource consumption breakdown for the last minute.

The following table describes the information displayed in the Database Overview view.

Table 5-5   Database Overview view

View area	Description
Session	Displays the following session-related information: ■ Sessions—number of sessions (different SPIDs) that were connected to the database during the last sample. ■ Active Sessions—number of sessions (different SPIDs) that were in one of the In MS-SQL states during the last sample. ■ Locked Sessions—number of blocked sessions connected to the database during the last sample.
Statistics	Displays information on the following statistical parameters: ■ Memory Usage (Max)—displays the maximum memory (in MB) consumed by all sessions that were connected to the selected database during the last 15 minutes. ■ Memory Usage (Total)—displays the total memory (in MB) consumed by all sessions that were opened in SQL Server during the last 15 minutes. ■ Open Transactions (Max)—displays the maximum transactions opened by all the sessions that were connected to the selected database during the last 15 minutes. ■ Open Transaction (Total)—displays the total transactions opened by all the sessions that were connected to the selected database during the last 15 minutes.

Table 5-5   Database Overview view

View area	Description
In MS-SQL (Last minute)	Displays the resource consumption breakdown of the database during the last minute (that is, 60 samples of the Collector). In addition to the resource distribution, the view displays the number of sessions in each state during the last sample.
	It is important to remember that since this view only displays the last minute's activities, you will always be viewing the activities that occurred during the last minute and not the activities that occurred since the last refresh action.
	The following information is displayed:
	■ State—displays the In MS-SQL state.
	■ In MS-SQL—graphical representation of the Time column.
	■ Time—amount of time the selected entity was in this state.
	■ %—percentage of time the selected entity was in this state.
	■ Sessions (Last Sample)—Current number of sessions in the specified state. For example, there are currently three sessions experiencing a Lock Wait state.
	Note: All the information displayed in this table represents information collected during the last minute, with the exception of the Sessions column, which represents the number of sessions, in the last sample, that were in a particular state.

See About session states.

About viewing Database locks

Displays lock wait duration broken down into lock levels such as Table, Page and Key, for the selected Database, during the last minute. The sessions that are currently involved in a locking or blocking situation are displayed in tree format.

Note: Since the Lock tree represents the locks that are currently occurring, and the lock breakdown represents the duration of locks occurring during the last minute, it is possible that a situation may occur whereby a lock was freed, but since the last minute has not yet passed, the lock tree will display the message 'No locks found' while the Lock Breakdown grid still shows lock duration.

The following table provides a detailed explanation of the Lock Tree and Lock Breakdown (Last Minute) mechanism.

Table 5-6   Database Locks view

View area	Description
Lock tree	It is important to determine the following information for each blocking chain:
	■ Which sessions are involved in the lock?
	■ Which session is blocking the others?
	■ Which object is blocked?
	■ What type of lock is causing the block?
	The Lock Tree provides the necessary information to get to the root of the problem by providing information on sessions currently involved in the blocking situation.
	A yellow lock indicates that the session is currently locking 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 itself.
	To gain a better understanding of the lock tree take a closer 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 Lock Tree displays the SPID, program name, the object being blocked and the lock type.

Table 5-6   Database Locks view

View area	Description
Lock breakdown (Last minute)	Displays the Lock Wait duration broken down into lock levels such as Table, Page and Key, for the selected database, during the last minute: ■ Lock Level—such as, Row, Key, Page, Table, or Other. "Other" includes all other locks. ■ Lock Wait—a graphical representation of the data displayed in the Time column. ■ Time—amount of time the selected database was locked at the specified lock level. ■ %—percentage of time the selected database was locked at the specified lock level. ■ Sessions (Last Sample)—current number of sessions locked at the specified lock level. **Note:** All the information displayed in this table represents information collected during the last minute, with the exception of the Sessions column, which represents the number of sessions, in the last sample, that were in a particular state.

See About session states.

About the Session entity

Displays information on a single connection with Microsoft SQL Server. Precise for SQL Server identifies a session by the following attributes:

- System Process ID (SPID)—a unique integer assigned to each user connection when the connection is made.

- Login time

- Session identifiers such as program, database user, or login

When one of these attributes changes, a new session is recorded and displayed.

The following table describes the information displayed in the session overview.

Table 5-7 Session Overview

View area	Description
Session Identifiers	Displays information on the session identifier's connection details, session-related statistics and a breakdown of session resource consumption, during the last minute.
	Session identifiers:
	State—current state of the session.
	Status—current status of the session as reported by SQL Server, such as:
	■ Sleeping—session is idle.
	■ Runnable—SPID is currently executing.
	■ Dormant—same as Sleeping.
	■ Rollback—session is rolling back the transaction.
	■ Defwakeup—session is waiting for a resource that is in the process of being freed. The waitresource field should indicate the resource in question.
	■ Spinloop—session is waiting while attempting to acquire a spinlock used for concurrent control on Symmetric Multi-Processing (SMP) systems. See *SQL Server Books Online* for more information.
	Login Time—time the session started.
	Last Batch Time—if the session is currently executing a batch this field displays the time the batch started; otherwise it displays the time the previous batch finished executing.
	Duration—time elapsed since login time.
	Idle Time—time elapsed since last batch.
	Open Transactions—number of open transactions for the SPID.
	Parallel Sessions—current number of subthreads used to execute a batch in parallel. This counter is calculated from the ecid column in sysprocesses. It enables you to verify that SQL Server uses the best execution plan for the current statement.
	Blocked By—if the session is blocked by another session, displays the SPID of the blocker session.
	Wait Type—an internal SQL Server field that signifies the current wait identifier, used by the Collector to assign a Session State.
	Wait Time—if the session is waiting this field displays the current wait time in milliseconds.

Table 5-7   Session Overview

View area	Description
Connection	A connection includes session identifiers, such as, database, login, machine, user and work type, as reported by sysprocesses.
	In ERP systems (if the appropriate ERP extension is installed), Precise for SQL Server replaces some identifiers as follows:
	In SAP:
	■ Login -> SAP User
	■ Work Type -> Dialog, Update, Spool, Background, Enqueue
	■ Machine -> User Application
	In PeopleSoft:
	■ Login -> PS User ID
	■ Work Type -> Interactive, Batch
	In Siebel:
	■ Login -> Siebel User
	■ Program -> Siebel View
	■ Machine -> Siebel Server
	■ Work Type -> Interactive, Batch, Background
	In COM+:
	■ Work Type -> COM+
Statistics	Displays information on the following statistical parameters:
	■ CPU Time— CPU time for the sessions during the last minute.
	■ Physical I/O Operations—total number of physical I/O requests performed by the session during the last minute.
	■ Memory Usage—number of pages in the procedure cache that are currently allocated to the SPID. A negative number indicates that the process is freeing memory allocated by another process.
In MS-SQL (last minute)	Displays the resource consumption breakdown of the session during the last minute (that is, 60 samples of the Collector). In addition to the resource distribution, the view displays the number of sessions in each state during the last sample.
	It is important to remember that since this view only displays the last minute's activities, you will always be viewing the activities that occurred during the last minute and not the activities that occurred since the last refresh action.
	The following information is displayed:
	■ State—displays the In MS-SQL state.
	■ MS-SQL—graphical representation of the Time column.
	■ Time—amount of time the selected entity was in this state.
	■ %—percentage of time the selected entity was in this state.
	Note: All the information displayed in this table represents information collected during the last minute, with the exception of the Sessions column, which represents the number of sessions, in the last sample, that were in a particular state.
	See <u>About session states</u>.

The following table describes the information displayed in the Session Locks view.

Table 5-8 Session Locks view

View area	Description
Locks	Displays lock wait duration broken down into lock levels such as Table, Page and Key, for the selected Session, during the last minute. The sessions that are currently involved in a locking or blocking situation with the selected session are displayed in tree format.
	Since the Lock tree represents the locks that are currently occurring, and the lock breakdown represents the duration of locks occurring during the last minute, it is possible that a situation may occur whereby a lock was freed, but since the last minute has not yet passed, the lock tree will display the message 'No locks found' while the Lock Breakdown grid still shows lock duration.
	The Lock tree provides the necessary information to determine the following information for each blocking chain.
	A yellow lock indicates that the session is currently locking 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 itself.
	To gain a better understanding of the lock tree take a closer 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 Lock Tree displays the SPID, program name, the object being blocked and the lock type.
	The Lock breakdown (Last minute) displays the Lock Wait duration broken down into lock levels such as Table, Page and Key, for the selected session, during the last minute:
	■ Lock Level—such as, Row, Key, Page, Table, or Other. "Other" includes all other locks.
	■ Lock Wait—A graphical representation of the data displayed in the Time column.
	■ Time—Amount of time the selected session was locked at the specified lock level.
	■ %—Percentage of time the selected session was locked at the specified lock level.
	Note: The table only displays lock levels that the selected session was blocked by. If the selected session was not blocked, the information area remains empty.

The following table describes the information displayed in the Session Text view.

Table 5-9 Session Text view

View area	Description
Text	Displays the text of the current statement. If the session is not active and executing a statement, a message notifying you that the session is not active and that no text is available is displayed.

About the Locked Object entity

The Overview view in the Main area displays information on the objects that are currently locked (such as, Database, Object, Sessions, and Lock Type).

The following table describes the information displayed in the Locked Object overview view.

Table 5-10 Locked Object Overview view

Column	Description
Database	Displays the database containing the blocked object.
Object	Displays the object being blocked.
Sessions	Displays the number of sessions that hold locks on the object or are blocked while trying to acquire a lock on the object.
Lock Type	Indicates the type of lock that is held against the object.

You can display information on Locked Objects, in the Association area, when the selected entity is an instance or database. If an instance is selected, the Association area shows information on all currently locked objects in all the databases in the instance. If a database is selected, the table shows specific information on all currently locked objects in the selected database. Moving the cursor over the icon in the first column displays a ToolTip that identifies the object's lock type.

The following table describes the information displayed the Association area table.

Table 5-11 Information displayed in Association area for Locked Objects

Column	Description
Icon	Indicates if there is a session waiting for the object.
Object	Displays the object being blocked
Sessions	Number of sessions that hold locks on the object or are blocked while trying to acquire a lock on the object.
Locks	Total number of locks, broken down into the following lock types: Exclusive, Update, Shared, Intent Exclusive, Intent Update, and Intent Shared.
Exclusive	Indicates how many of the locks placed on the sessions were exclusive locks and shows a breakdown of the lock modes into lock levels, such as, Table, Row, Key, and Page. When exclusive locks are placed on a resource, the holding session is granted exclusive access to that resource and it can perform modification operations.
Update	Indicates how many of the locks placed on the sessions were update locks and shows a breakdown of the lock modes into lock levels, such as, Table, Row, Key, and Page. Update locks are used to prevent deadlocks. A deadlock occurs when more than one session intends to update a resource and tries to lock the resource for future potential updates. The update lock is an interim stage which is promoted to an exclusive lock when the update is actually made.
Shared	Indicates how many of the locks placed on the sessions were shared locks and shows a breakdown of the lock modes into lock levels, such as, Table, Row, Key, and Page. When there are shared locks on an object, concurrent transactions can read but not modify the resource.
Intent Exclusive	Indicates how many of the locks placed on the sessions were intent exclusive locks and shows a breakdown of the lock modes into lock levels, such as, Table, Row, Key, and Page. Indicates that a transaction intends to modify some subordinate resources in the lock hierarchy by placing exclusive locks on those particular resources.

Table 5-11 Information displayed in Association area for Locked Objects

Column	Description
Intent Update	Indicates how many of the locks placed on the sessions were intent update locks and shows a breakdown of the lock modes into lock levels, such as, Table, Row, Key, and Page.
	Indicates that a transaction intends to update some subordinate resources in the lock hierarchy, at a later stage, by placing update locks on those particular resources.
Intent Shared	Indicates how many of the locks placed on the sessions were intent shared locks and shows a breakdown of the lock modes into lock levels, such as, Table, Row, Key, and Page.
	Indicates that a transaction intends to read but not modify some subordinate resources in the lock hierarchy by placing shared locks on those particular resources.

About the Session Holding Locks entity

The fact that a currently active session is holding a resource does not necessarily mean that there is a contention between resources. The Session Holding Locks entity displays information that lets you differentiate between sessions that are locking resources and sessions that are holding resources.

The following views display information on sessions holding locks:

- Overview
- Locks
- Text

About getting an overview of current sessions holding locks

Displays information on the session identifier, including connection details, session-related statistics and a breakdown of session resource consumption, during the last minute.

The following table describes the information displayed by the Session Holding Locks overview.

Table 5-12   Sessions Holding Locks overview

View area	Description
Session Identifiers	Displays the following information on the session's identifiers: State—current state of the session. Status—current status of the session as reported by SQL Server, such as: ■ Sleeping—session is idle. ■ Runnable—SPID is currently executing. ■ Dormant—same as Sleeping. ■ Rollback—session is rolling back the transaction. ■ Defwakeup—session is waiting for a resource that is in the process of being freed. The waitresource field should indicate the resource in question. ■ Spinloop—session is waiting while attempting to acquire a spinlock used for concurrent control on Symmetric Multi-Processing (SMP) systems. See *SQL Server Books Online* for more information. Login Time—time the session started. Last Batch Time—if the session is currently executing a batch this field displays the time the batch started; otherwise it displays the time the previous batch finished executing. Duration—time elapsed since login time. Open Transactions—number of open transactions for the SPID. Parallel Sessions—current number of subthreads used to execute a batch in parallel. This counter is calculated from the ecid column in sysprocesses. It enables you to verify that SQL Server uses the best execution plan for the current statement. Blocked by—if the session is blocked by another session, displays the SPID of the blocker session. Wait Type—an internal SQL Server field that signifies the current wait identifier, used by the Collector to assign a Session State. Wait Time—if the session is waiting this field displays the current wait time in milliseconds.
Connection	A connection includes session identifiers, such as, database, login, machine, user and work type, as reported by sysprocesses. In ERP systems (if the appropriate ERP extension is installed), Precise for SQL Server replaces some identifiers as follows: In SAP: ■ Login -> SAP User ■ Work Type -> Dialog, Update, Spool, Background, Enqueue ■ Machine -> User Application In PeopleSoft: ■ Login -> PS User ID ■ Work Type -> Interactive, Batch In Siebel: ■ Login -> Siebel User ■ Program -> Siebel View ■ Machine -> Siebel Server ■ Work Type -> Interactive, Batch, Background In COM+: ■ Work Type -> COM+

Table 5-12 Sessions Holding Locks overview

View area	Description
Statistics	Displays information on the following statistical parameters: ■ CPU Time— CPU time for the sessions during the last minute. ■ Physical I/O Operations—total number of physical I/O requests performed by the session during the last minute. ■ Memory Usage—number of pages in the procedure cache that are currently allocated to the SPID. A negative number indicates that the process is freeing memory allocated by another process.
In MS-SQL (last minute)	Displays the resource consumption breakdown of the session during the last minute (that is, 60 samples of the Collector). In addition to the resource distribution, the view displays the number of sessions in each state during the last sample. It is important to remember that since this view only displays the last minute's activities, you will always be viewing the activities that occurred during the last minute and not the activities that occurred since the last refresh action. The following information is displayed: ■ State—displays the In MS-SQL state. ■ MS-SQL—graphical representation of the Time column. ■ Time—amount of time the selected entity was in this state. ■ %—percentage of time the selected entity was in this state. **Note:** All the information displayed in this table represents information collected during the last minute, with the exception of the Sessions column, which represents the number of sessions, in the last sample, that were in a particular state. See About session states.

About viewing information on the lock breakdown of a session that is holding locks

The Locks view displays lock wait duration broken down into lock levels such as Table, Page and Key, for the selected Session, during the last minute.

The sessions that are currently involved in a locking or blocking situation with the selected session are displayed in tree format. Since the Lock tree represents the locks that are currently occurring, and the lock breakdown represents the duration of locks occurring during the last minute, it is possible that a situation may occur whereby a lock was freed, but since the last minute has not yet passed, the lock tree will display the message 'No locks found' while the Lock Breakdown grid still shows lock duration.

To gain a better understanding of the lock tree take a closer 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 Lock Tree displays the SPID, program name, the object being blocked and the lock type.

The following table describes the information displayed in this view.

Table 5-13 Locks view for the Session Holding Locks entity

View area	Description
Lock Tree	The Lock tree provides the necessary information to determine the following information for each blocking chain: ■ A yellow lock indicates that the session is currently locking 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 itself. ■

Table 5-13 Locks view for the Session Holding Locks entity

View area	Description
Lock breakdown (Last Minute)	The Lock breakdown (Last minute) displays the Lock Wait duration broken down into lock levels such as Table, Page and Key, for the selected session, during the last minute:
	■ Lock Level—such as, Row, Key, Page, Table, or Other. "Other" includes all other locks.
	■ Lock Wait—A graphical representation of the data displayed in the Time column.
	■ Time—Amount of time the selected session was locked at the specified lock level.
	■ %—Percentage of time the selected session was locked at the specified lock level.
	Note: The table only displays lock levels that the selected session was blocked by. If the selected session was not blocked, the information area remains empty.

About viewing the text of a session that is holding locks

Displays the text of the current statement. If the session is not active and executing a statement, a message notifying you that the session is not active and that no text is available is displayed.

About Statement and Batch entities

The Statement and Batch entities display information on a SQL statement that was executed by the selected session during the last time slice.

About getting an overview of current Statement and Batch entities

Displays general information on the statement or batch, including overall properties, its breakdown In MS-SQL during the last time slice, and the breakdown of the locks during the last time slice.

When the selected entity is a Batch, the information area displays the entire text of the ad hoc batch or stored procedure.

When the selected entity is a Statement, the information area displays only the part of the batch text that forms the statement.

The following table describes the information displayed in the Statement or Batch Overview view.

Table 5-14 Statement or Batch Overview view

View area	Description
Statement/Batch ID	Displays the internal statement hash value (SHV) calculated by the Collector, or, in the case of a stored procedure, displays the name of the stored procedure.
Database	Identifies the database on which the statement or batch is run. This information is used by Precise for SQL Server to display the statement's execution plan.
User	Displays the parsing user who runs the statement or batch. This information is used by Precise for SQL Server to display the statement's execution plan.
Executions	Number of times the Collector identifies that the statement or batch was executed.
Duration	Total time required to execute the selected statement or batch during the last time slice.
Duration (Avg)	Average duration of each statement or batch execution during the last time slice.
CPU Time	Cumulative CPU time for the statement or batch during the last time slice.
Physical I/O Operations	Total number of physical I/O requests during the execution of the statement or batch during the last time slice.

Table 5-14 Statement or Batch Overview view

View area	Description
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.
In MS-SQL (Last minute)	Displays the resource consumption breakdown of the statement or batch during the last minute (that is, 60 samples of the Collector). In addition to the resource distribution, the view displays the number of sessions in each state during the last sample.
	It is important to remember that since this view only displays the last minute's activities, you will always be viewing the activities that occurred during the last minute and not the activities that occurred since the last refresh action.
	The following information is displayed:
	■ State—displays the In MS-SQL state.
	■ In MS-SQL—graphical representation of the Time column.
	■ Time—amount of time the selected entity was in this state.
	■ %—percentage of time the selected entity was in this state.

About viewing current Statement and Batch text

Displays text of the selected statement or batch.

How the Current 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
- Observing current session activity
- Examining a single connection
- Viewing ERP data
- Examining contentions
- Analyzing locks currently held
- Monitoring tempdb usage

Examining resource consumption of an entire instance and database

The Current tab provides an almost real-time picture of the performance and behavior of your SQL Server instance (or part of your instance). The Current tab displays information on the last time slice of activity, unless otherwise configured as for the last minute. Usually you will examine the activities of the entire instance level to determine what is happening in your system. By examining the current activity at an instance level of display, you can see how it is currently behaving and determine if there is an overall problem that affects all sessions or if there is a problem with specific sessions.

- See Examining contentions.

To examine resource consumption of an entire instance and database

1   In the Instance list, choose the instance you want to analyze.
     Do one of the following:

- On the View controls in the Main area, click **Overview**. This will show you a breakdown of resource consumption for the last minute.

- On the View controls, click **Statistics** to analyze the current statistics of the selected instance.

2 Launch to the Statistics tab, in-context, to view additional statistical information on the selected instance.

Observing current session activity

The Association area of the Current tab lets you view which sessions are currently connected to the Instance, or which sessions are currently using a specific database or holding a specific object. You can observe the current state of the session and determine if it is active or waiting for a resource. You can also identify the session by different identifiers such as its login information, the machine it is running on, or the programs it is running. The currently active TSQL code is displayed for active sessions.

To observe current session activity

1 In the Instance list, choose the instance you want to analyze.

2 On the Association controls in the Association area, select one of the following to focus your investigation on the sessions that were active in SQL Server during the last minute:

- Active Sessions

- Sessions

- Sessions Holding Locks

Examining a single connection

You may need to explore a single running SPID, to obtain more information on it. In the Association area you can drill down to the specific SPID in question, and continue your analysis.

To examine a single connection

1 In the Instance list, choose the instance you want to analyze.

2 In the Association area, drill down to the session whose connection you want to analyze.

3 Continue your analysis by examining its resource breakdown, its recent statements, or the batches in which the session was executed, during the last few minutes.

4 On the View controls, click **Locks** and observe which locks the session is holding or waiting for.

Viewing ERP data

Certain obstacles make it difficult to investigate ERP sessions. Most activity is registered in SQL Server under generic names. For example, in PeopleSoft®, all PeopleSoft activities are listed under a single login (sa) and generic program names (PeopleSoft).

Installing the Precise for SQL Server Interpoint extension enables you to correlate SQL Server information with ERP information. For example, if you install Interpoint for Siebel, you will be able to correlate between SQL Server sessions to Siebel.

To view ERP data

1 Verify that the Precise for SQL Server Interpoint extension is installed.
   For more information, see the *Precise Installation Guide*.

2 In the Instance list, choose the instance you want to analyze.
   For this example, you are viewing Siebel session data.

3 In the Association area, view Siebel session data for each session in the selected instance and observe to which SQL Server session it is correlated to.

Examining contentions

One of the common uses of the Current tab is to solve contentions. A contention is a situation in which a session waits for a resource held by another session. In a busy application, there can be situations in which one session locks many other sessions that are waiting for the resource. For example, every time a user logs on to the system the Logon table is updated. If the system is not properly tuned a situation could arise where while the Logon table is being held by another session no user could log in into the system. This type of scenario can be resolved very easily using Precise for SQL Server. You can observe all the locks on the instance, in the Locks view of the Instance entity.

To examine contentions

1   In the Instance list, choose the instance you want to analyze.

2   On the View controls in the Main area, click **Locks** and examine the locking contentions.

3   Check which sessions and programs are holding the locks and locate which objects are involved in contentions.

4   In the Association area, on the Locks tab, observe the status of the locking session and the type of lock (for example, table lock or page lock).

See Examining locking over time of a database entity.

Analyzing locks currently held

To be able to avoid a blocking situation, it is important to have an accurate picture of all the locks held in the system. This will help you understand if the lock system is tuned and if it is possible to avoid a potential blocking situation. Using Precise for SQL Server, you can now observe the locks currently held in the Instance and group them according to lock level and lock mode.

To analyze locks currently held

1   In the Instance list, choose the item you want to analyze.

2   On the View controls in the Main area, click **Locks** and examine the locking contentions. Move the cursor over the locked session in the tree to display the text of the locked session.

3   On the Association controls, in the Association area, click **Locked Objects**, and examine the locks in SQL Server. Identify which objects have many locks held on them, which objects are in use, and how they are being held.

4   Examine all the locks in the instance shows which objects are in use (there is a lock on the object) and how they being held. As you can see the table is involved in a contention with other sessions.

5   If you find an object that is locked, examine the sessions that are holding locks on the object by drilling down to the object.

6   On the View controls in the Main area, click **Overview** and view the lock type.

7   After drilling down to the object, on the Association controls in the Association area, click **Sessions** to find the sessions that hold locks on the object.

Monitoring tempdb usage

Exploring Tempdb usage is very important during the tuning and monitoring process. Even a well-tuned application can suddenly suffer from degradation in response time, due to a session that was busy with the Tempdb database. It is therefore important to monitor Tempdb performance.

To monitor Tempdb usage

1   In the Instance list, choose the instance you want to analyze.

2   On the View controls in the Main area, click **Tempdb Usage** and analyze Tempdb performance.

Examining 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.

Note: 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.

Note: 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 6-1 Precise for SQL Server Activity tab structure
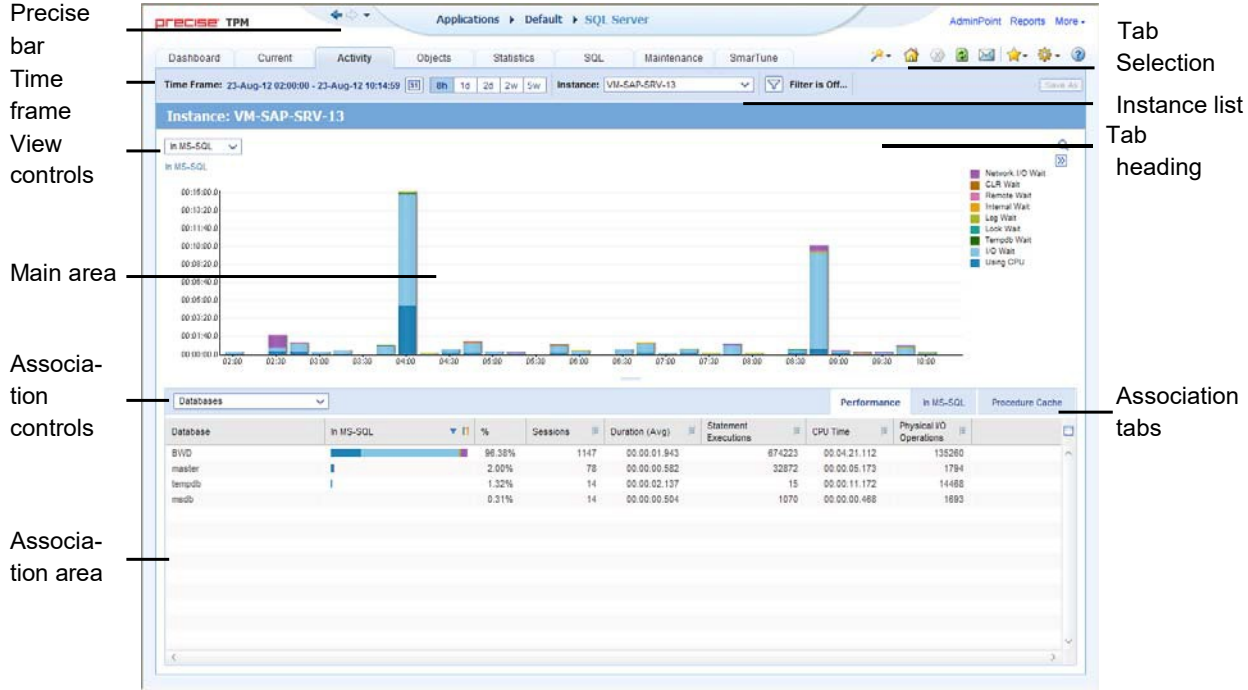
The Activity tab displays information on various entities and their associated entities. The information displayed when you launch the Activity tab depends upon where you are launching from.

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 the following topics for more information:

- [About drilling down in context](#).

- [About the Precise bar](#).

- [About the Main area](#).

- [About the Association area](#).

- [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 the following topics for more information:

- [About drilling down in context](#).
- [About the Precise bar](#).
- [About the Main area](#).
- [About the Association area](#).
- [Focusing on information in overtime graphs](#).
- [How most tabs are structured](#)
- [Tasks common to most tabs](#)
- [About the Dashboard tab](#)

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

Note: 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:
	Note: 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.

Note: 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 6-1 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" on page 36.

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.

Note: 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 6-2   Information displayed in the Performance tab

Column	Description
Entity Name (such as Instance, Database)	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 6-3   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.

Table 6-3   Information displayed in the Procedure Cache tab

Column	Description
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 6-4   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 will display "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.

Note: 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 adhoc 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
	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 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 increases.
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 <u>Focusing on information in overtime graphs</u>.

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 6-5   Information displayed in the Performance tab

Column	Description
TUNE	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 6-6 Information displayed in the Statistics tab

Column	Description
TUNE	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 <command>ecid</command> 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 6-7 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 6-8   Information displayed in the Plan tab

Columns	Description
TUNE	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).
Nested 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 6-9   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 6-10   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 `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 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 6-11 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.

Note: 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.

Note: 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.

Note: 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 6-12   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.

Table 6-12 Information displayed in the Performance tab

Column	Description
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 6-13 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 6-14 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.

Table 6-14   Lock types

Type	Description
Unknown	Precise for SQL Server is not able to identify the lock type.

Note: 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 an 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.

Note: 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.

Note: 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 6-15 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.

Table 6-15   Benefits of the Precise for SQL Server Interpoint extension

Type of application	Benefit
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.

Note: 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

Since 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.

Note: 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.

Examining the relationship between database objects

This section includes the following topics:

About the Objects tab

It is important to understand the relationship between database objects and statements to make well-founded tuning decisions.

The Objects tab is a comprehensive browser that allows you to understand the relationships and associations between database schema objects. It also enables further understanding of the relationships between data dictionary components and the statements that access them. This information allows you to identify the statements that will be affected by changes to a database object, and the schema relationship impacts that are associated with an object change.

The Objects tab is also very useful for determining scaling strategies for your database. Are you considering scaling out or scaling up? If you thoroughly understand the relationship between a table, the SQL executed on the table and the resources consumed by these SQL statements, you can identify the tables that may be candidates for either a horizontal or vertical scale-out. To engage a successful scale-out, you need to ensure that the tables you are horizontally partitioning to different databases engage in enough activity and resource consumption to spread the load across multiple systems. The same is true for vertical scaling. If you know the overall resource consumption of the instance and the consumption per table, you can make educated decisions as to how to best vertically partition your tables into multiple databases.

The Objects tab is also very useful for scale-ups. 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 system and resolve them. Only once the major wait states are resolved, will a scale-up actually be able to improve performance. Otherwise, you may be simply wasting money and valuable resources.

The Objects tab also allows you to observe storage utilization, over time, at varying levels, such as, database, file group, database files.

While the Activity tab focuses on SQL tuning, the Object tab focuses on database object tuning. In many cases, you may want to tune a database object rather than a specific query or program. This is especially true if you did not personally write the application in question, as for example in ERP and CRM applications, and cannot therefore change the text of any statement.

Object tuning is a powerful tool. Taking into consideration important aspects regarding objects can ensure maximum application performance. Precise for SQL Server allows you to examine your index structure by displaying the usage of each index. Storage layout can be optimized by analyzing the object storage configuration and metric information.

Since Precise for SQL Server tracks changes in the database objects, the effect of schema changes on performance can be easily observed.

How the Objects tab is structured

The Objects tab displays information on a selected entity and its associated entities. When you open the Objects tab the selected entity is by default Instance, meaning that information is displayed on the instance level. When you launch the Objects tab from the Activity tab with one of the following entities—Database, Table, Index, DB file or Locked Object—the tab is launched in context with the entity you selected in the Activity tab.

If you open the Objects tab from another tab, the historical settings (those settings which were selected when you left the tab, such as the last entity you drilled down to) are taken into account and the information displayed the last time you viewed the tab is displayed (similar to clicking the History button and returning to a previous tab).

The selected entity is always reflected in the Tab heading, which serves as a point of orientation. The highest level entity you can view information for in the Objects tab is the Instance level. You can select an instance from the Instance list.

The entities displayed in the Association area are associated with the selected entity displayed in the Main area. At times, the relationship between the entity displayed in the Main area and those displayed in the Association area is that of parent to child and sometimes it merely represents that there is a relationship between the selected entity and the entities displayed in the Association area.

About the Main area in the Explore Objects tab

The Main area shows comprehensive information on the selected entity. You can choose from several views to examine the entity from different angles. You can, for example, focus exclusively on SQL Server Log data, or Instance Parameters, or Instance Parameters changes.

About the Association area in the Explore Objects tab

The Association area provides corresponding information on the child entities. You can view information on one type of child entity at a time, such as instances only or databases only, by selecting a control from the View Controls list. The selection you make is reflected in the Association area only; the Main area remains unchanged. For some entities, tabs above the Association area enable you to view additional information: Clicking a tab changes the table columns for the same type of child entities.

From the Association area, you can also drill down to another entity by clicking a table row. A drilldown affects the whole tab: When you drill down to another entity, the Tab heading reflects the new selection, the Main area displays information on the newly selected entity, and the Association area shows data on its associated entities.

For example, when you want to view information on a specific logical file, choose Logical Files from the Association controls. The Association area changes to display all the files groups associated with the selected Logical File. Note that the Tab heading and the Main area remain unchanged. In the Association area, click the row of the logical file you want to view detailed information for. The Tab heading indicates the newly selected entity; the Main area displays an overview of the logical file you want to investigate, and the Association area shows general details on the file groups that are associated with the Logical File you selected.

See About Precise for SQL Server tabs.

About the Recommend tab

Precise for SQL Server uses the Microsoft® Index Tuning Wizard to achieve recommended indexes or statistics for the selected table. This helps the optimizer choose a better access plan and enhances the performance of the statements or batches that access the selected table. The recommendations are based on a workload. The workload contains all the statements stored in the PMDB that were executed during the selected time period that have an average duration time exceeding the value defined in the registry (where 0 is the default). The recommend indexes process only performs the recommend process on indexes or statistics additions.

The following information is displayed in the Recommend tab:

- Recommended Index/Statistics

- Recommendation details

About viewing Recommended Indexes/Statistics

Displays the recommended indexes or statistics, as follows:

- Icon—indicates whether the item is a recommended index or statistic.

- Table—displays the name of the table to which the recommendation refers.

- Unique—indicates whether the index is unique. This value is not relevant for recommended statistics.

- Clustered—indicates whether the index is clustered. This value is not relevant for recommended statistics.

- Columns List—displays a list of the index's or statistic's columns separated by commas. The sort order (ASC, DESC) is also displayed for recommended indexes.

About viewing Recommendation details

Displays details on the recommended index or statistic.

The following tabs are available:

Columns	Displays a list of all the columns in the recommended index's or statistic's table as follows:
	■ Icon—indicates whether the data in the column will be stored in the index in ascending (ASC) or descending (DESC) order. No icon is displayed for columns that are not part of the recommended index.
	■ Column—name of the column.
	■ Key Number—key number of the column in the index.
	■ Column Type—the physical storage type of the column.
	■ Part of Index—indicates whether the column participates in an existing index.
	■ First Key of Index—indicates whether the column participates as the first key in an existing index.
	■ Used—number of times the column is used in the breakdown of the workload statements according to the following types: Range Scan, Table Scan, Sort and Join.
DDL	Displays the DDL text of the recommended index or statistic.
Table's Indexes/Statistics	Displays a list of the existing indexes or statistics of the table selected in the recommendation area.
	The following information is displayed:
	■ Icon—indicates whether the item is an index or statistic.
	■ Name—displays the name of the index or statistic.
	■ Unique—indicates whether the index is unique. This value is not relevant for statistics.
	■ Clustered—indicates whether the index is clustered. This value is not relevant for statistics.
	■ Columns List—displays a list of the columns of the index or statistic separated by commas. The sort order (ascending, descending) is also displayed for indexes.
Analyzed Statements	Displays a list of all the statements in the workload.
	The following information is displayed:
	■ Statement—indicates the statement ID.
	■ In MS-SQL—displays the resource consumption (as a stacked bar) of the specified statement within the selected time period.
	■ Text—displays the statement's text.

About Objects tab entities

The following entities can be examined in the Objects tab:

- [Instance](#)
- [Database](#)
- [Active Objects](#)
- [Table and Table Contain Same Column](#)
- [Index and Unused Index](#)
- [Partition](#)
- [Column](#)
- [View and View Contain Same Column](#)
- [User](#)
- [Logical File](#)
- [File Group](#)
- [Statistics](#)
- [Stored Procedure](#)
- [Function](#)
- [Trigger](#)
- [Statement](#)
- [Batch](#)
- [Physical Disk](#)
- [Logical Volume](#)
- [Disk Partition](#)
- [About the Storage Unit entity](#)

About the Instance entity

The Instance entity displays configuration information, configuration changes, as collected by the Collect Schema Changes process, and performance information, as collected by the Precise for SQL Server Collector for the entire instance.

The following views are available:

- Overview
- SQL Server Log
- Instance Parameters
- Instance/Database Changes
- Databases Created/Dropped

About getting an overview of instance configuration and performance information

The Overview displays configuration and performance information for the entire instance. The Overview is divided into the following areas:

Server Details	Displays server configuration details such as version and number of CPUs.
Instance Details	Displays SQL Server instance configuration details such as version, service pack and server details.
In MS-SQL	Displays resource consumption, over the selected time period, of the selected instance, as collected by the Precise for SQL Server Collector.

Space Allocation	Displays the total free space and total used space (in MB) over the selected time period, of all the files in the selected instance, as collected by the Collect Space Utilization process.

About viewing SQL server log messages

The SQL Server Log view displays the SQL Server log messages from the recent error log file, for the selected time period.

About viewing instance configuration parameters

The Instance Parameter view displays the selected instance's configuration parameters. This includes the possible range values and configuration values, for each parameter. A change icon is displayed if the Collect Schema Change Process detects a change in any of the parameters.

Precise for SQL Server reports changes that occurred within the selected time period. If you observe that a parameter was changed, select Instance Parameter Changes from the View controls, to view additional information.

About viewing changes to instance parameters and database options

The Instance/Database view displays the changes to instance parameters and database options vs. resource consumption, over the selected time period, to enable you to identify configuration performance changes caused by these actions, as follows:

Instance/Database Changes	Sample Date—the date when the changes were captured by the Collect Schema Changes process.
	Database—the name of the database where the change occurred. For Instance Configuration changes, the value is N/A.
	Data Changed—displays one of the following values:
	■ Schema—any change in the database schema.
	■ Database option—a description of the database option that was changed
	■ Configuration Parameter—a description of the instance configuration parameter that was changed.
	New Value—displays the new value that resulted following the change.
	Old Value—displays the previous value that preceded the change.
In MS-SQL	Displays resource consumption of the selected instance, over the selected time period, as collected by the Precise for SQL Server Collector.
Schema Changes	Displays an overtime graph that indicates whether changes occurred in the database, or a breakdown of the instances into the following groups:
	■ Schema Changed—any change in one of the database's Schema.
	■ Database Option Changed—any change in one of the database's options.
	■ Instance Property Changed—any change in one of the instance configuration parameters.

The following configuration parameters that can be displayed:

- Maximum recovery interval in minutes

- Number of locks for all users

- Number of open database objects

- Maximum worker threads.

- Network packet size

- Memory for index create sorts (KB)

- Priority boost

- Minimum memory per query (KB)

- Query wait(s)

- Set working set size

- Affinity mask

- Cost threshold for parallelism

- Maximum degree of parallelism

- Minimum Server Memory size (MB)

- Maximum Server Memory size (MB)

- Maximum estimated cost of query allowed to run by query governor

- User mode scheduler uses lightweight pooling

- Recovery Model Changed

- Auto Update Statistics Changed

- Ansi Null Changed

- Auto Shrink File Changed

- Close Cursors On Commit Changed

- Status Changed

- Auto Create Statistics Changed

- Truncate Log Changed

- Compatibility Level Changed

About viewing databases that were created or dropped

The Databases Created/Dropped view displays the databases that were dropped or created vs. resource consumption over the selected time period to enable you to identify performance changes caused by these actions.

In MS-SQL	Displays resource consumption of the selected instance, over the selected time period, as collected by the Precise for SQL Server Collector.
Schema Changes	Displays resource consumption of the selected instance, over the selected time period, as collected by the Precise for SQL Server Collector.

About the Database entity

The Database entity displays configuration information, configuration changes collected by the Collect Schema Changes process and performance information collected by the Precise for SQL Server collector of the selected database.

The following views are available:

- Overview

- Database Options

- Database Option Changes

- Object Changes

About getting an overview of configuration, performance and storage information for a selected database

The Overview displays configuration, performance and storage information for the selected database, as follows:

Properties	Displays general database details, such as, status, collation, and compatibility level.

In MS-SQL	Displays resource consumption of the selected database, over the selected time period, as collected by the Precise for SQL Server Collector.
Space Allocation	Displays total free space and total used space (in MB), over the selected time period, of all the files in the selected database, as collected by the Collect Space Utilization process.
Schema Changes	Displays an overtime graph that indicates whether changes were made to the database objects and the database configuration options.
Data vs. Log	Displays total free space and total used space (in MB), over the selected time period, of all the files in the selected database, as collected by the Collect Space Utilization process, and displayed according to file types as datafiles or log files.

About viewing database options

The Database Options view displays the database options, such as Auto Create Statistics and Auto Shrink File, etc. If the Collect Schema Changes process detects a change in any of the options, an icon is displayed in the first column. Select the Database Option Changes view to observe additional information regarding the detected change. The following information is displayed:

Database Options	Displays the current value of the database options.
In MS-SQL	Displays resource consumption information of the selected database, over the selected time period. Schema Changes Displays option changes of the selected database, over the selected time period, as detected by the Schema Change Process.

About viewing database option changes

The Database Option Changes view displays the database option changes vs. resource consumption, over the selected time period, so as to enable you to identify configuration changes that led to performance problems.

The following database options changes are displayed:

- Recovery Model Changed
- Ansi Null Changed
- Close Cursors On Commit Changed
- Auto Create Statistics Changed
- Compatibility Level Changed
- Auto Update Statistics Changed
- Auto Shrink File Changed
- Status Changed
- Truncate Log Changed

The following information is displayed:

Database Option Changes	Displays the changes made to the database options of the selected database, over the selected time period.
In MS-SQL	Displays resource consumption information of the selected database, over the selected time period.
Schema Changes	Displays option changes of the selected database, over the selected time period, as detected by the Schema Change Process.

About viewing object changes

The Object Changes view displays the schema changes vs. resource consumption over the selected time period to enable you to identify performance changes caused by these actions, as follows:

Object Changes	Sample Date—the estimated last change date of the table (and /or its sub-objects). The change is reported by the Precise for SQL Server Schema Change Process, which runs, by default, once a day. This means that the Sample Date actually represents the date when the Schema Change Process was executed.
	Object Type—the Object type can be one of the following: Table, View, Stored procedure, Trigger, Foreign key, Scalar function, Inline Table function, Table function, and Extended storage procedure.
	Object—displays the object name.
	Object Owner—displays the owner of the object.
	Change Type—can be one of the following:
	■ Created—indicates object was created.
	■ Dropped—indicates object was dropped.
	■ Index—indicates any change that was made in the object's index(es).
	■ Change in structure—indicates any change that was made in the object's column(s) and the creation or deletion of the object's trigger(s).
	■ Change in properties—indicates any change that was made in one of the following object's properties: Name, Owner, File Group and Exec is ANSI null on.
	■ Partition function
	■ Partition schema
	■ Partition function values
In MS-SQL	Displays resource consumption of the selected database, over the selected time period, as collected by the Precise for SQL Server Collector.
Schema Changes	Displays an overtime graph that indicates whether changes were made to the database objects, broken down into the following groups:
	■ Index—indicates any change, addition, or deletion of an index.
	■ Object Change in Structure—indicates any change in the object's structure. For example, whether a new column was added to a table.
	■ Object Change in Property—indicates any change in the object's property. For example, whether the name of the object was changed.
	■ Object Created/Dropped—indicates whether any of the objects sampled (table, view, stored procedure, etc.) was dropped or created.

About the Database entity in the Association area

You can view the entities associated with the selected Database entity in the Association area. It is possible to associate to the Database entity from the Instance entity.

About the Active Objects entity

The Active Objects entity pinpoints objects (tables including their indexes statistics) that may be a good candidate for tuning. If the object is an SQL Server 2005, additional information regarding I/O operations and waits, contentions and logical operations is also displayed. Clicking on the object 's icon launches to the Objects tab with the selected object in context.

The Active Objects entity indicates the following:

- the name of the user, object, and database it is associated with

- a summary of the time it spent in SQL Server, during the time frame you selected

- its size (in pages)

See About the Table and Table Contain Same Column entities and About the Index and Unused Index entities.

About the Active Objects Association area

The following additional information is available for SQL Server 2005 objects:

I/O Operations	Indicates the amount of I/O operations performed on the table or index, over the selected time period.
I/O Waits	Displays the I/O wait for the table or index, over the selected time period.
Contentions	Displays the number of contentions experienced by the table or index, over the selected time period.
Contention Waits	Displays the contention wait data for the selected table or index, over the selected time period.
Logical Operations	Displays how many range scan and Singleton lookups were performed on the table or index, over the selected time period.
Size (pages)	Displays the size of the table or index (in pages).

About the Table and Table Contain Same Column entities

The Table entity displays general details regarding resource consumption, space utilization and schema changes for the selected table.

The Table Contain same Column Entity displays tables that have the same column as the column displayed in the Main area. The information displayed is the same as displayed in the Table entity and can only be associated to from the Column entity.

The following views are available:

- Overview
- Show Contig
- Change Log Summary
- Change Log Detailed
- Usage Report
- Common Access Patterns (SQL Server 2005 only)
- Operational Statistics (SQL Server 2005 only)

About getting an overview of table entities

The Overview displays general table properties, along with the resource consumption, space utilization, schema changes and number of rows in the selected table, over the selected time period.

Properties	Displays general table details, such as, number of columns, file group, and whether or not the table uses a clustered index.
In MS-SQL	Displays the Using CPU and I/O Wait counters of the selected table, over the selected time period, based on its contribution to overall performance, when accessed by the execution plan.
Space Allocation	Displays the amount of free space and used space (in MB) in the selected table, over the selected time period, and specifies whether the space is populated by an index or data.
	The values may be incorrect due to out-of-date space usage information. To update the values, use the `DBCC UPDATEUSAGE` command to recalculate the space usage fields. For more information, see *SQL Server Books Online.*

Schema Changes	Displays an overtime graph that indicates whether changes were made to the selected table and a breakdown of its indexes or columns into the following groups:

- Table Change in Property—indicates any change made to one of the table's properties, such as if the table was renamed.
- Table Change in Structure—indicates any change made to the table's structure, such as whether a new column was added to the table.
- Index Change in Property—indicates any change made to one of the index's properties, such as whether the index was changed from clustered to non-clustered.
- Index Change in Structure—indicates any change made to the index's structure, such as whether a new key was added to the index.
- Index Created/Dropped—indicates any addition or deletion of indexes.

Rows	Displays the number of rows in the selected table, over the selected time period, as collected by the Collect Space Utilization process.

About viewing data storage and fragmentation information in the Show Contig view

The Show Contig view displays data storage and fragmentation information for the selected table.

The following information is displayed:

Minimum Record Size	Indicates the minimum record size in the data pages.
Maximum Record Size	Indicates the maximum record size in the data pages.
Average Record Size	Indicates the average record size in the data pages.
Number of Forwarded Records	When there is not enough free space to write an updated record in its page, SQL Server allocates the updated record to another page and instructs the old record to point to the new location.
	The Number of forwarded records is the number of records allocated to another page, since there was no free space in the original page.
	Accessing forwarded records adversely affects performance because additional I/O is performed. Check the number of forwarded records in your table or index in relation to the total number of records. If the ratio is high, consider reorganizing the table or index to achieve a better performance. To reorganize an index, use `DBCC DBREINDEX.` If the index is a clustered index, a Reindex action will reorganize the data pages.
Pages	Displays the number of data pages.
Extents	Displays the number of extents used to store the data pages.
Extent Switches	The number of times SQL Server moved from one extent to another while scanning the data pages (not including the first extent access).
Scan Density	Indicates the ratio between the optimum number of extent switches and the actual number of extent switches. This value determines the general level of fragmentation in a table. The higher the value, the better. The scan density is not valid, if the table spans multiple files.
Best Count	The optimum number of extent switches. This number can be achieved, if the extents and pages are perfectly linked, and there is no fragmentation.
Actual Count	Number of times the SQL Server moved from one extent to another while scanning the data pages (including the first extent access).
Average Free Bytes per Page	Average free bytes per page in the data pages.

Average Page Density (%)	The average page density is a percentage indicating how full the data pages are. Space that cannot be utilized due to row size constraints is counted as used. The higher the percentage, the better.
	When space is available within the index pages, that is, when the indexes are not making the most efficient use of space, scanning the table involves more read operations than if no free space were available on the pages. However, keeping free space on the pages is sometimes desirable to accommodate future expansion in the storage of the table's data and to reduce the potential for page splits. You can request to keep free space available by specifying a low fill factor value, when you create an index. For example, a fill factor of "10" means that the pages will be 10% full while creating the index. Having room on a page means that there is space to insert more rows without having to split a page. Splitting is a relatively expensive operation and can lead to a break of contiguity between pages, because the new page is not usually contiguous to the page being split.
Logical Fragmentation (%)	The logical fragmentation indicates how well ordered the data pages are inside an extent and between extents.
	"Well ordered" means that the number of the current page scanned is higher than the previous one. The logical fragmentation is computed by dividing the number of out-of-order pages by the number of the data pages. The lower the value, the less fragmentation there is. This is relevant only when there is a clustered index on the table, because ordered pages are important.
Extent Fragmentation (%)	The extent fragmentation indicates how well the extents, which contain the data pages, are physically contiguous on the disk.
	The page number of its first page identifies an extent. If the extents 8, 16, 24, or 32 make up an index, then we can see that the extents are contiguous on the disks, which means there are no gaps. On the other hand, if the extents 8, 24, 32, and 64 make up an index, then we can see that we have a gap. The extent fragmentation is computed by dividing the number of gaps by the number of extents, so in this example, the extent fragmentation is 1/4, or 25 percent.
	The lower the value, the less fragmentation there is.
	This number is not relevant to heaps, because physical order has no importance in the data pages.

Note: By default, not all dbcc fields are displayed in the Show Contig view.

About viewing changes made to the table and index levels in the Change Log Summary

The Change Log Summary view displays changes made to the table and index levels vs. resource consumption, over a selected time period, to enable you to locate schema changes that caused performance problems. The following information is displayed:

Change Log Summary	Sample Date—indicates the estimated last change date of the table (and/or its sub-objects). This value reflects the date that the change was first identified by the Precise for SQL Server Schema Changes Process, not the date the object was changed.
	Object Type—indicates whether the object is a table or index.
	Object—displays the table or index name.
	Change Type—Indicates the type of change detected. The following change types are reported:
	■ Created—indicates that the index was created
	■ Dropped—indicates that the index was dropped
	■ Change in Structure Table—indicates any change that was made to the table's column(s) and the creation or deletion of the table's trigger(s).
	■ Index—indicates any change that was made to the index keys of the index or to the index depth.
	■ Change in Properties Table—indicates any change that was made to one of the following table's properties: Name, Owner or File Group.
	■ Index—indicates any change that was made to one of the following index's properties: File Group, Fill Factor, Is Auto Statistics, Is Clustered Index, Is Pad Index, Is Page Lock Disallowed and Is Row Lock Disallowed.

In MS-SQL	Displays resource consumption of the selected table, over a selected time period, based on its contribution to overall performance, when accessed by the execution plan.
Schema Changes	Displays an overtime graph that indicates whether changes were made to the selected table and a breakdown of its indexes or columns into the following groups: ■ Table Change in Property—indicates any change made to one of the table's properties, such as if the table was renamed. ■ Table Change in Structure—indicates any change made to the table's structure, such as whether a new column was added to the table. ■ Index Change in Property—indicates any change made to one of the index's properties, such as whether the index was changed from clustered to non-clustered. ■ Index Change in Structure—indicates any change made to the table's index properties, such as whether a new key was added to the index. ■ Index Created/Dropped—indicates any addition or deletion of indexes.

About viewing schema changes made on a selected table in the Change Log Detailed view

The Change Log Detailed view details the schema changes made on the selected table and its sub-objects (column, index and index key).

The following table information is monitored by the schema change process:

- Name
- File Group
- Exec is ANSI null on
- Table has Cluster Index
- Insert Trigger Count
- Owner
- No. of Columns
- Table has Index
- Delete Trigger Count
- Update Trigger Count
- Partition Schema
- Partition Function

About viewing usage information and access methods in the Usage Report

The Usage Report view displays usage information and access methods based on statements stored in the PMDB. The following information is displayed:

Report was run with the following parameters	Displays usage information of the selected index, as follows: ■ Start Date—Indicates the minimum execution date for a statement, within the selected time period. ■ End Date—Indicates the maximum execution date for a statement, within the selected time period.

Statement Explain	Displays the following items: ■ Statement Access the Object—indicates the total number of statements that were explained successfully in the PMDB that refer to the selected table, within the selected time period. ■ Statement Not Access the Object—indicates the total number of statements that were explained successfully in the PMDB that do not refer to the selected table, within the selected time period. ■ Statements Not Explained Yet—indicates the total number of statements in the PMDB that were not yet explained, within the selected time period. ■ Statements Explained Unsuccessfully—indicates the total number of statements in the PMDB that were explained unsuccessfully, within the selected time period.
Table Profile	Displays a breakdown, by statement type, of the number of statements that referred to the specified table (SELECT, INSERT, UPDATE, DELETE); for example, 100 statements performed SELECT on the specified table. The In MS-SQL counters (Using CPU and I/O Wait) are also displayed.

About viewing common access patterns for tables or index (SQL Server 2005 only)

The Common Access Patterns view displays information that indicates to what extent a table or index was used. The following information is displayed:

User Operations	Displays the number of user seeks, scans, lookups or updates.
System Operations	Displays the number of system seeks, scans, lookups or updates.
User vs. System Operations	Graph comparing user operations to system operations, over the selected time period.
Last User or System Seek, Lookups, Scans or Updates	Displays when the last user or system seek, lookup, scan or update operation occurred.

About viewing operational statistics

The Operational Statistics view displays information that indicates whether or not an index or table is being used. The following graphs are displayed:

- Range scan vs. Singleton lookups
- Physical I/O operations
- Locking and Latching contentions
- Locking and Latching contention waits

About the Table and Table Contain Same Column entities Association area

You can view general details of the entities associated with the selected Table entity in the Association area. It is possible to associate to the Table entity from the following entities: Database, Column, Index, Physical Disk, Logical Volume, HP Storage Device, HDS Storage Device, Statement, Batch, Stored Procedure, Function and Trigger.

The following additional information is available for SQL Server 2005 objects:

Partitions	Indicates the number of partitions defined for the selected table.
Table Partitioned	Indicates whether or not the table is partitioned.
Partition Schema	Indicates under which file groups to map the partition.
Partition Function	Displays the name of the function that defines how the table should be partitioned.

About the Index and Unused Index entities

The Index entity displays general details regarding resource consumption, space utilization and schema changes for the selected index.

The Unused Index entity displays general details regarding space utilization and schema changes for the selected index. An index that is not in use is identified through the Execution plan of the statements stored in the PMDB (SQL Server 200) or through the relative dynamic views (DMV) supplied by SQL Server 2005.

The following views are available:

- Overview
- Show Contig
- Statistics
- Schema Changes
- Usage Report
- Common Access Patterns (SQL Server 2005 only)
- Operational Statistics (SQL Server 2005 only)

About getting an overview of space utilization and schema changes for a selected index

The Overview displays space utilization and schema changes for a selected index, as follows:

Properties	Displays general details on the index such as number of keys, fill factor, and whether the index is clustered or unique.
In MS-SQL	Displays Using CPU and I/O Wait of the selected index over a selected time period, based on its contribution to overall performance when accessed by the execution plan.
Space Allocation	Displays the amount of free space and used space (in MB) in the selected index, over the selected time period.
	Note: The values may be incorrect due to out-of-date space usage information. To update the values, use the DBCC UPDATEUSAGE command to recalculate the space usage fields. For more information, see *SQL Server Books Online.*
Schema Changes	Displays an overtime graph that indicates whether there where index changes, broken down into the following groups: ■ Change in Property—indicates any change made to the index properties, such as if the index was changed from clustered to non-clustered. ■ Change in Structure—indicates any change made to the index structure, such as if a new key was added to the index.
Rows	Displays the number of rows in the selected index, over the selected time period, as collected by the Collect Space Utilization process.

About viewing data storage and fragmentation information of a selected index in the Show Contig view

The Show Contig view displays data storage and fragmentation information for the selected index.

The following information is displayed:

Minimum Record Size	Displays the minimum record size of the pages in the leaf level of the index.
Maximum Record Size	Displays the maximum record size of the pages in the leaf level of the index.
Average Record Size	Displays the average record size of the pages in the leaf level of the index.

Number of Forwarded Records	When there is not enough free space to write an updated record in its page, SQL Server allocates the updated record to another page and instructs the old record to point to the new location.
	The Number of forwarded records is the number of records allocated to another page, because there was no free space available in the original page.
	Every access to a forwarded record requires an additional step than access to a regular record, which affects performance. Check the number of forwarded records in your table or index in relation to the total number of records. If the ratio is high, consider reorganizing the table or index to achieve better performance. Use `DBCC DBREINDEX` to reorganize an index. If the index is a clustered index, reindexing will reorganize the data pages.
Pages	Displays the number of pages in the leaf level of the index.
Number of Extents	Number of extents used to store the pages in the leaf level of the index.
Extent Switches	Displays the number of times SQL server moved from one extent to another while scanning the leaf level pages (not including the first extent access).
Scan Density	Indicates the ratio between the optimum number of extent switches and the actual number of extent switches. This value determines the general level of fragmentation in an index. The higher the value, the less fragmentation there is in the index. The scan density is not valid if the table spans multiple files.
Best Count	Displays the optimum number of extent switches. This number can be achieved, if the extents and pages are perfectly linked, and there is no fragmentation.
Actual Count	Displays the number of times SQL Server moved from one extent to another while scanning the index pages (including the first extent access).
Average Free Bytes per Page	Displays the average free bytes per page in the leaf level pages.
Average Page Density (%)	The average page density is a percentage indicating how full the index pages are.
	Space that cannot be utilized due to row size constraints is counted as used. The higher the percentage, the better.
	When space is available within the index pages, that is, when the indexes are not making the most efficient use of space, this means that scanning the table involves more read operations than if no free space were available on the pages. However, maintaining free space on the pages is sometimes desirable to accommodate future expansion in the storage of the table's data and to reduce the potential for page splits. You can request to keep free space available by specifying a low fill factor value, when you create an index. For example, a fill factor of "10" means that the pages will be 10% full while creating the index. Having room on a page means that there is space to insert more rows without having to split a page. Splitting is a relatively expensive operation and can lead to a break of contiguity between pages, because usually the new page is not contiguous to the page being split.
Logical Fragmentation	Indicates how well ordered the leaf pages are inside an extent and between extents. Well ordered means that the number of the current page scanned is higher than the previous one. Logical fragmentation is computed by dividing the number of out-of-order pages by the number of pages in the table. The lower the value, the less fragmentation there is in the index. This is relevant only when there is a clustered index on the table because then there is significance to ordered pages.
Extent Fragmentation (%)	The extent fragmentation indicates how well the extents, which contain the leaf pages, are physically contiguous on the disk.
	The page number of its first page identifies an extent. If the extents 8, 16, 24, or 32 make up an index, then we can see that the extents are contiguous on the disks, which means there are no gaps. On the other hand, if the extents 8, 24, 32, and 64 make up an index, then we can see that we have a gap. The extent fragmentation is computed by dividing the number of gaps by the number of extents, so in this example, the extent fragmentation is 1/4, or 25 percent.
	The lower the number, the less fragmentation there is.
	This number is not relevant to heaps, because physical order has no importance in the data pages.

Note: By default, not all `dbcc` fields are displayed in the Show Contig view.

About viewing distribution statistics for a selected index

The Statistics view displays distribution statistics for the selected index.

General

Displays general information on the distribution statistics, as follows:

Statistics Updated	Indicates the last time the statistics were updated.
Row Sampled	Indicates the number of rows sampled for the statistics calculation.
Average Key Length	Indicates the average length of all the keys in the index.
Steps	The number of steps that were sampled and used in the statistics calculation. A "step" is a range of key values (the first key in the index) that were sampled.

The following values are recorded for each step:
- High value in the range
- Density = 1 / (1+No. of distinct values in the range)
- No. of rows in the range (not including the High value)
- No. of rows equal to the "High value"
- No. of distinct values in the range (not including the "High value")
- Avg. No. of rows per distinct value ("No. Of rows in range" / "No. of distinct values in the range")

Rows Sampled	The percentage of rows sampled for the statistics calculation.

About Density

Indicates how selective an index is, as follows:

Density	This indicator provides a measure of how selective the index is. The more selective an index is, the more useful it is, since this means that more rows can be eliminated from consideration. For example, a unique index is the most selective index since each index entry can point to only one row.
	Density = 1/No. of unique values. Density values can be between "0" and "1". The lower the Density value, the more selective the index.
Average Length	Indicates the average length of the specified set of keys.
Columns	The left-based subset of key columns in the index (including the first column alone). If there is a clustered index on the table, its keys are considered part of the key in every non-clustered index row, because non-clustered index rows point to the clustered index rows. For example, if there is a non-clustered index with the columns, "CustomerID" and "OrderDate". The table also contains a clustered index, "OrderID".

The columns will be:
- CustomerID
- CustomerID, OrderDate
- CustomerID, OrderDate, OrderID

About Distribution

Displays additional distribution statistics for the selected index, as follows:

High Value	Each row in the Distribution table represents information about a range of key values (the first key in the index). This range is called a "Step". "High value" is the upper value in the range.
Rows in Range	Indicates the number of rows in the sample that have the specified key range values, not including the high value in the range.

Rows Equal to High Value	Indicates the number of rows from the sample that have the same value as the specified High value
Distinct Values in Range	Indicates the number of distinct values inside the range not including the High value.
Rows per Distinct Values in Range (Avg)	The average number of rows, per distinct value, in the specified range, not including the High value in the range. This value is computed by dividing the number of rows in the range by the number of distinct values in the range (the number of distinct values in the range always being greater than zero).
Density	Step Density means the uniqueness of the data values. This information provides a measure of how selective the index is. The more selective an index is, the more useful it is, because this means more rows can be eliminated from consideration. A unique index, for example, is the most selective index, where each index entry can point to only one row.
	Step Density = 1/(1+No. of distinct values in the range) In
	this case, we calculate the density of the current step.
	Step Density values can be between "0" and "1". The lower the Step Density value, the more unique the data values.

About viewing schema changes for a selected index and index key

The Schema Changes view shows a table that summarizes the schema changes made on the selected index and index keys. For example, this would include whether the first key of an index was dropped, making the index unusable in statements.

The following index information is monitored by the schema change process:

- File Group
- Fill Factor
- Is Pad Index
- Index Depth
- Is Page Lock Disallowed
- No of Keys
- Is Clustered Index
- Is Unique
- Is Auto Statistics
- Is Row Lock Disallowed
- Position
- Order by
- Partition Schema
- Partition Function

About viewing a usage report

The Usage Report view displays usage information and access methods based on statements stored in the PMDB.

The following information is displayed:

Report was run with the following parameters	Displays usage information of the selected index, as follows: ■ Start Date—indicates the minimum execution date for a statement, within the selected time period. ■ End Date—indicates the maximum execution date for a statement, within the selected time period.

Statement Explain	Displays statement explain, as follows: ■ Statement Access the Object—indicates the total number of statements that were explained successfully in the PMDB that refer to the selected index, within the selected time period. ■ Statement Not Access the Object—indicates the total number of statements that were explained successfully in the PMDB that do not refer to the selected index, within the selected time period. ■ Statements Not Explained Yet—indicates the total number of statements in the PMDB that were not yet explained, within the selected time period. ■ Statements Explained Unsuccessfully—indicates the total number of statements in the PMDB that were explained unsuccessfully, within the selected time period.
Table Profile	Displays a breakdown, by statement type, of the number of statements that referred to the specified table (SELECT, INSERT, UPDATE, DELETE); for example, 100 statements performed SELECT on the specified table. The In MS-SQL counters (Using CPU and I/O Wait) are also displayed.

About viewing common access patterns for indexes (SQL Server 2005 only)

The Common Access Patterns view displays information that indicates to what extent an index was used. The following information is displayed:

User Operations	Displays the number of user seeks, scans, lookups or updates.
System Operations	Displays the number of system seeks, scans, lookups or updates.
User vs. System Operations	Graph comparing user operations to system operations, over the selected time period.
Last User or System Seek, Lookups, Scans or Updates	Displays when the last user or system seek, lookup, scan or update operation occurred.

About viewing operational statistics

The Operational Statistics view displays information that indicates whether or not the index on a table is being used. The following graphs are displayed:

- Range scan vs. Singleton lookups
- Physical I/O operations
- Locking and Latching contentions
- Locking and Latching contention waits

About the Index and Unused Index entities Association area

You can view general details on the entities associated with the selected Index entity in the Association area. It is possible to associate to the Index entity from the following entities: Database, Table, Column, View, User, File Group, Physical Disk, Logical Volume, HP Storage Device, HDS Storage Device, Statement and Batch.

The following additional information is available for SQL Server 2005 objects:

Partitions	Indicates the number of partitions defined for the selected index.
Index Partitioned	Indicates whether or not the index is partitioned.
Partition Schema	Indicates under which file groups to map the partition.
Partition Function	Displays the name of the function that defines how the index should be partitioned.
Index Aligned	Indicates whether or not the index is aligned with its table.
Number of Non-key Columns	Displays the number of non-key columns used within an index.
Index Non-Key Columns	Displays the non-key column names.

About the Partition entity (SQL Server 2005 only)

The Partition entity displays information on how an index or table is partitioned. All tables and indexes in a database are considered to be partitioned, in SQL Server 2005, even if they are made up of only one partition. The data of partitioned tables and indexes is divided into units that can be spread across more than one file group in a database. The data is partitioned horizontally, so that groups of rows are mapped into individual partitions.

The following view is available:

- Overview

About getting an overview of an index or table that is partitioned

The Overview displays general details on the selected index or table, as follows:

Properties	The following information is displayed for each partition:
	■ Object
	  Name of the object being partitioned.
	■ Partition
	  Displays the partitions id.
	■ Filigreed
	  Indicates where the data related to a partition is stored.
	■ Rows
	  Displays the number of rows per partition.
	■ Max Value
	  Indicates the upper value of a partition.
	■ Min Value
	  Indicates the lower value of a partition.
	■ Upper Boundary
	  Indicates whether data falling on the maximum value will be assigned to the next partition or to the same partition—yes will assign it to the next partition; no to the same partition it falls in.

Space Allocation	Displays the amount of space utilized by the partitions.
Rows	Displays the number of rows in a partition, over the selected time period.

About the Partitions entity Association area

You can view general details on the entities that are associated with the selected Partition entity in the Association area. It is possible to associate to the Partition entity from an Index or Table entity.

About the Column entity

The Column entity displays general details and schema changes for the selected column.

The following views are available:

- Overview
- Schema Changes

About getting an overview of a selected column

The Overview displays general details on the selected column such as type, length, and default value.

About viewing schema changes made on a selected column

The Schema Changes view shows a table that summarizes the schema changes made on the selected column, such as, if the column length has changed.

The following column information is monitored by the schema change process:

- Name

- Length

- Precision

- Type

- Scale

- Is Nullable

About the Column entity Association area

You can view general details on the entities that are associated with the selected Column entity in the Association area. It is possible to associate to the Column entity from the Table, Index, and View entities.

SQL Server 2005 allows you to extend the functionality of non-clustered indexes by adding non-key columns to the leaf level of the non-clustered index. This allows you to create non-clustered indexes that cover more queries, thereby considerably improving query performance. The Non-Key Column field (displayed for SQL Server 2005 non-clustered indexes only), indicates whether or not the selected column is a non-key column.

About the View and View Contain Same Column entities

The View entity displays general details and schema changes for the selected view.

The View Contain Same Column entity displays views that have the same column as the column displayed in the Main area. The information displayed is the same as displayed in the View entity and can only be associated to from the Column entity.

The following views are available:

- Overview

- Change Log Summary

- Change Log Detailed

About getting an overview of the View and View Contain Same Column entities

The Overview displays general details on the view such as creation date, number of columns and displays the view text.

About viewing the Change Log Summary of the selected view and its indexes

The Change Log Summary displays schema changes made in the selected view and its indexes, as follows:

Change Log Summary Table	The following information is displayed:
	Sample Date—the estimated last change date of the view (or/and its sub-objects). This value reflects the date that the change was first identified by the Precise for SQL Server Schema Changes Process, not the date that the object was changed.
	Object Type—indicates whether the object type is a View or Index.
	Object—displays the name of the View Index.
	Change Type—indicates the type of change detected. The following change types are reported: ■ Created—indicates that the index was created ■ Dropped—indicates that the index was dropped ■ Change in Structure View: indicates any change that was made to the view's column(s). Index: indicates any change that was made to the index keys of the index or to the index depth. ■ Change in Properties View: indicates any change that was made to one of the following view's properties: Name and Owner. Index: indicates any change that was made to one of the following index's properties: File Group, Fill Factor, Is Auto Statistics, Is Clustered Index, Is Pad Index, Is Page Lock Disallowed and Is Row Lock Disallowed.
Schema Changes	Displays an overtime graph that indicates whether changes were made to the selected view and its indexes or columns, broken down into the following groups: ■ View Change in Property—indicates any change made to the view properties. For example, whether the view was renamed. ■ View Change in Structure—indicates any change made to the view structure. For example, whether a new column was added to the view. ■ Index Change in Property—indicates any change made to one of the view's index properties, such as, the index was changed from clustered to non-clustered. ■ Index Change in Structure—indicates any change made to the index structure, for example, whether a new key was added to the index. ■ Index Created/Dropped—indicates any addition or deletion of indexes.

About viewing Change Log Details

The Change Log Detailed view displays the schema changes made on the selected view and/or its sub-objects.

About the View and View Contain Same Column entities Association area

You can display general details on the entities that can be associated with the selected View entity in the Association area. It is possible to associate to the View entity from the Database, Table, Column and Index entities.

About the User entity

The User entity displays general details of the selected user, such as, Roles and Database Access.

About the User entity Association area

You can view general details on the entities that can be associated to the selected User entity in the Association area. It is possible to associate to the User entity from the Database entity.

About the Logical File entity

The Logical File entity displays general details, I/O activity and storage information on the selected database file.

The following views are available:

- Overview
- Statistics

About getting an overview of the Logical File entity

The Overview displays general details on the Logical File, as follows:

Properties	Displays general details on the Logical File such as full filename (including path), file group, and file growth.
I/O Wait	Displays the I/O wait for the selected Logical File, over the selected time period, as collected by the Collect DB File Statistics process.
I/O Read Wait	Displays the I/O read wait for the selected Logical File, over the selected time period, as collected by the Collect DB File Statistics process.
I/O Write Wait	Displays the I/O write wait for the selected Logical File, over the selected time period, as collected by the Collect DB File Statistics process.
Bytes Requested	Displays the I/O activity (bytes read and written) over the selected time period, for the selected Logical File, as collected by the Collect DB File Statistics process.
Requests	Displays the I/O activity (number of read/writes) over the selected time period, for the selected Logical File, as collected by the Collect DB File Statistics process.
Space Allocation	Displays free space and used space (in MB) over the selected time period, for the selected logical file, as collected by the Collect Space Utilization process.

About viewing I/O statistics on a Logical File

The Statistics view displays I/O statistics on the Logical File over the selected time period, as follows:

I/O Wait	Displays the I/O wait for the selected Logical File, over the selected time period, as collected by the Collect DB File Statistics process.
Bytes Requested	Displays the I/O activity (bytes read and written) over the selected time period, for the selected Logical File, as collected by the Collect DB File Statistics process.
Requests	Displays the I/O activity (number of read/writes) over the selected time period, for the selected Logical File, as collected by the Collect DB File Statistics process.

About the Logical File entity Association area

You can view general details on the entities associated to the Logical File entity in the Overview tab in the Association area. The Statistics tab shows information on I/O wait, bytes requested, number of requests, and size of logical files. It is possible to associate to the Logical File entity from the following entities: Instance, Database, File Group, Physical Disk, Logical Volume, HP Storage Device, HDS Storage Device, Symmetrix Storage Device and Clariion Storage Device.

About the File Group entity

The File Group entity displays general details and space utilization, over the selected time period, for the selected file group.

About the Overview

The Overview displays general File Group details such as whether the file group is read-only and information regarding the file group's space utilization, over the selected time period, as follows:

Properties	Displays general File Group details such as whether the file group is read only, or whether this is the default file group.
Storage	Displays the amount of free space and used space of the selected file group, over the selected time period, based on the data collected by the Collect Space Utilization process.

About the File Group entity Association area

You can view general details on the entities that are associated to the selected File Group in the Association area. It is possible to associate to the File Group from the following entities: Database, Table, Index, Logical File, Physical Disk, Logical Volume, HP Storage Device and HDS Storage Device.

About the Statistics entity

The statistic entity displays distribution statistics on the selected collection (set of columns).

The following views are available:

- Overview
- Statistics

About getting an overview of the selected statistic

The Overview displays general details of the selected Statistic such as keys and an indication when the statistics were last calculated.

About viewing distribution statistics

The Statistics view displays distribution statistics for the selected statistic.

See About viewing distribution statistics for a selected index.

About the Statistics entity Association area

You can view general details on the selected Statistics entity in the Association area. It is possible to associate to the Statistics entity from the Table and Column entities.

About the Stored Procedure entity

The stored procedure entity displays general details and resource consumption of the selected stored procedure.

The following views are available:

- Overview
- Text

About getting an overview of a stored procedure

The Overview displays general properties of the stored procedure, provides resource consumption information, in addition to the number of times the selected stored procedure's statements were executed, based on the data collected by the Precise for SQL Server Collector for the selected time period.

The following information is displayed:

Properties	Displays general details of the stored procedure such as whether the stored procedure is an extended procedure, and whether the stored procedure was created during installation.

In MS-SQL	Displays the resource consumption of the selected stored procedure, based on the data collected by the Precise for SQL Server Collector for the selected time period.
Attributes	Displays selected attributes that exist for each stored procedure, such as if the stored procedure has the flag `Anis Null On` set to *On* or *Off*.
Statement Execution	Displays the number of times the selected stored procedure's statements were executed, based on the data collected by the Precise for SQL Server Collector for the selected time period.

About viewing the text of a stored procedure

The Text view displays the stored procedure's text.

About the Stored Procedure entity Association area

You can view general details on the entities associated with the selected Stored Procedure entity in the Association area. It is possible to associate to the Stored Procedure entity from the Database, Table, View, User, Stored Procedure and Function entities.

About the Function entity

The function entity displays general details and resource consumption of the selected function.

The following views are available:

- Overview
- Text

About getting an overview of the Function entity

The Overview displays general details of Function entity, as follows:

Properties	Displays general properties of the function, such as type of function (inline, scalar, etc.), and whether or not the function was created during installation.
In MS-SQL	Displays the resource consumption of the selected function, based on the data collected by the Precise for SQL Server Collector for the selected time period.
Attributes	Displays selected attributes that exist for each stored procedure, such as if the stored procedure has the flag `Anis Null On` set to *On* or *Off*.
Statement Executions	Displays the number of executions of the selected function's statements, based on the data collected by the Precise for SQL Server Collector for the selected time period.

About viewing the text of a function

The Text view displays the text of the function.

About the Function entity Association area

You can view general details on the entities associated with the selected Function entity in the Association area. It is possible to associate to the Function entity from the Database, Table, View, User, Stored Procedure and Function entities.

About the Trigger entity

The trigger entity displays general details and resource consumption of the selected trigger.

The following views are available:

- Overview
- Text

About getting an overview of general trigger properties

The following information is displayed in the Overview:

Trigger	Displays general trigger properties such as trigger name, and the name of the table on which the trigger is defined, and whether this is the first trigger fired when data is deleted from a table.
In MS-SQL	Displays resource consumption of the selected trigger, over a selected time period, based on the data on trigger statements collected by the Precise for SQL Server Collector.
Attributes	Displays selected attributes that exist for each stored procedure, such as if the stored procedure has the flag `Anis Null On` set to *On* or *Off*.
Statement Executions	Displays the number of times the selected trigger's statements were executed, based on the data collected by the Precise for SQL Server Collector for the selected time period.

About viewing the text of a trigger

The Text view displays the text of the trigger.

About the Trigger entity Association area

You can view general details on the entities associated with the selected Trigger entity in the Association area. It is possible to associate to the Trigger entity from the Database, Table, View and User entities.

About the Statement entity

The Statement entity displays access plan details and resource consumption for the selected statement.

The following views are available:

- Overview
- Performance
- SQL Text

About getting an overview of a statement

The Overview displays general details about the statement, along with details regarding its last access plan, as follows:

Statement	Displays general statement details such as statement ID, database and parsing user.
Access Plan	Provides details regarding to the last access plan of the statement, such as whether a table scan, index seek and index scan was performed.

About viewing the performance of a selected statement

The Performance view displays resource consumption information on the selected statement, as follows:

In MS-SQL	Displays resource consumption information for the selected statement, over the selected time period.

Statement Executions   Displays the number of statement executions carried out over the selected time period.

About viewing the SQL text of a selected SQL statement

The SQL Text view displays the text of the selected SQL statement.

About the Statement entity Association area

You can view general details on the entities associated with the selected Statement entity in the Association area. It is possible to associate to the Statement entity from the Table and Index entities.

The following tabs are available provide additional information in the Association area:

Performance   Provides general details and resource consumption of the statement.

Plan   Displays details on the last access plan of the statement such as whether a Table Scan was used, and whether an Index Seek was.

About the Batch entity

The Batch entity displays access plan details and resource consumption for the selected batch.

The following views are available:

- Overview
- Performance
- SQL Text

About getting a general overview of batch details

The Overview displays general batch details such as last show plan and most recent plan, in addition to details regarding the selected batch's last access plan, as follows:

Batch   Displays general details of the batch such as database and parsing user.

Access Plan   Displays details on the last access plan of the batch such as, whether a table scan or index scan was performed.

About viewing the performance of a batch

The Performance view displays resource consumption and number of statement executions for a statement, as follows:

In MS-SQL   Displays resource consumption information for the selected statement, over the selected time period.

Statement Executions   Displays the number of statement executions carried out over the selected time period.

About viewing the SQL text of a batch

The SQL Text view displays the batch's text.

About the Batch entity Association area

You can view general details on the entities associated with the selected Batch entity in the Association area. It is possible to associate to the Batch entity from the Table and Index entities.

The following tabs are available in the Association area:

Performance   Displays general details and resource consumption for the selected batch.

Plan   Displays details on the last access plan of the selected batch such as whether a table scan or index seek was used.

About the Physical Disk entity

The Physical Disk entity displays general details and I/O wait data for the selected physical disk.

About getting an overview of general physical disk details

The Overview displays general physical disk details along with I/O wait data for the selected physical disk, as follows:

Properties	Displays general physical disk details such as type, and number of logical volumes.
I/O Wait	Displays the I/O wait on the selected physical disk as collected by Precise for SQL Server Collector over the specified time period.

About the Physical Disk entity Association area

You can view general details on the entities associated with the selected Physical Disk entity in the Association area. It is possible to associate to the Physical Disk entity from the following entities: Database, Table, Index, Logical File, File Group, Logical Volume, Disk Partition, HP Storage Device and HDS Storage Device.

About the Logical Volume entity

The Logical Volume entity displays general details and storage utilization for the selected logical volume.

About getting an overview of logical volume details

The Overview displays general logical volume details, in addition to the free space and used space of the selected logical volume, as collected by the Collect Space Utilization process over the specified time period, as follows:

Properties	Displays general details of the logical volume such as type, and number of Logical files stored in the logical volume.
Storage	Displays the free space and used space of the selected logical volume as collected by the Collect Space Utilization process over the specified time period.

About the Logical Volume entity Association area

You can view general details on the entities associated to the selected Logical Volume entity in the Association area. It is possible to associate to the Logical Volume entity from the following entities: database, table, index, logical file, file group, physical disk, disk partition, HP storage device and HDS storage device.

About the Disk Partition entity

The Disk Partition entity displays general details for the selected disk partition, such as disk partition number in physical disk and disk partition number in logical volume, disk partition offset and length.

About the Disk Partition entity Association area

You can view general details on the entities associated to the selected Disk Partition entity in the Association area. It is possible to associate to the Disk Partition entity from the Physical Disk, Logical Volume, HP Storage Device and HDS Storage Device entities.

About the Storage Unit entity

The Storage Unit entity displays a list of all storage entities connected to instances in the Tier. The Storage Unit entity can only be associated to from the Instance level.

For more information about the Storage Device entity, see About Storage Unit entities.

About the Storage Unit entity Association area

You can view general details on the entities associated to the selected Storage Units in the Association area. This information is based on the mapping carried out by the Precise for SQL Server Storage agents that are installed.

How the Objects tab can help you identify performance problems

The following section describes a few examples on how you can use the Objects tab to identify performance problems.

- Analyzing how configuration changes affect resource consumption
- Analyzing how schema changes affect the resource consumption of various entities
- Examining an object's space usage over time
- Examining fragmentation of table data and indexes
- Examining access methods to tables and indexes
- Examining the distribution statistics of indexes
- Determining whether to recommend an index for a table

Analyzing how configuration changes affect resource consumption

Changing an instance configuration parameter (or database parameter) may affect the performance of your system. Precise for SQL Server allows you to track the configuration changes and compare them with the resource consumption of your application, over time. This can be done by examining the Instance/Database Changes view in the Instance entity. The Instance/Database Changes view allows you to compare the changes made to the instance configuration parameters and the database options, with the resource consumption of the entire instance, over a selected time period. In this way you can pinpoint the changes that affect the resource consumption of the entire database.

To analyze how configuration changes affect resource consumption

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

2 In the Instance list, choose the item you want to analyze.

3 On the View controls in the Main area, click `Instance/Database Changes`. Compare the changes made to the instance configuration parameters and the database options, with the resource consumption of the entire instance, over the selected time period. This allows you to pinpoint the changes that affect the resource consumption of the entire database.

4 On the Association controls in the Association area, click `Databases`.

5 Compare the changes made to the instance configuration parameters and the database options, with the resource consumption of the entire instance, over the selected time period. This allows you to pinpoint the changes that affect the resource consumption of the entire database.

6 On the View controls in the Main area, click `Database Created/Dropped`. Analyze how changes to database options affect In MS-SQL resource consumption.

Analyzing how schema changes affect the resource consumption of various entities

Changing the database schema may affect the performance of your system. These changes are done to improve the performance of your application. You need to monitor them to ensure that you have achieved your goals. Precise for

SQL Server allows you track the schema changes and compare them with the resource consumption of the entire database or a table. This allows you to detect changes that affect the resource consumption of the database. The Object Changes view of the Database entity allows you to check the database objects that where changed, dropped or created in reference to the resource consumption of the entire database over time. A summary of the schema changes is displayed. Drilling down to the specific object provides detailed information on the schema changes.

See [About the Database entity](#).

To analyze how schema changes affect the resource consumption of various entities

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

2   In the Instance list, choose the item you want to analyze.

3   On the Association controls in the Association area, click **Databases**. Drill down to the database you want to analyze.

4   On the View controls in the Main area, click **Object Changes**. Check the database objects that where changed, dropped or created in reference to the resource consumption of the entire database over time and determine if they affect In MS-SQL resource consumption.

5   Drill down to a specific table entity. On the View controls in the Main area, click **Change Log Detailed** and view detailed information on the changes made to the table's schema.

Examining an object's space usage over time

Space usage information, collected by the Collect Space Utilization process, allows you to examine the space allocated versus the space used, in a table and index. This enables you to easily keep track of object growth over time and decide how to spread database files over various disks and how to connect objects to file groups in the specified database.

To examine an object's space usage over time

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

2   In the Instance list, choose the item you want to analyze.

3   On the Association controls in the Association area, click **Databases**. Drill down to the database you want to analyze.

4   Drill down to a specific table entity. On the View controls in the Main area, click **Overview**.

5   In the Main area, analyze the Space over Time graph to determine how tablespace has been utilized over time.

Note: The values might be incorrect due to out-of-date space usage information. To update the values, use the DBCC UPDATEUSAGE command to recalculate the space usage fields. For more information, see *SQL Server Books Online for DBCC UPDATE USAGE.*

Examining fragmentation of table data and indexes

Fragmentation of table data and indexes may affect the performance of your system. The fragmentation occurs as a result of data modification (INSERT, UPDATE, and DELETE statements) performed on the table. Fragmentation increases the amount of page reads performed by the query, increasing their resource consumption.

To examine fragmentation of table data and indexes

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

2   In the Instance list, choose the item you want to analyze.

3   On the Association controls in the Association area, click **Databases**. Drill down to the database you want to analyze.

4   On the Association controls in the Association area, click **Tables** or **Indexes**. Drill down to a specific table or index entity (depending upon your selection).

5   On the View controls in the Main area, click **Show Contig** and analyze the fragmentation information displayed for the selected table or index. Determine whether the table or index is heavily fragmented and needs to be reorganized.

See [About viewing data storage and fragmentation information in the Show Contig view](#) and [About viewing data storage and fragmentation information of a selected index in the Show Contig view](#).

Examining access methods to tables and indexes

Examining the access methods used on tables and indexes can help you make decisions regarding the types of indexes you should employ. For example if most of the statements accessing the specified table are of type update (INSERT, UPDATE, DELETE) and only a few are of type SELECT you should add as few indexes as possible to efficiently access data, on the one hand, and avoid overhead when performing updates, on the other hand. Also if you observe an increase in I/O wait you should consider rewriting the statements using the specified table or index, making sure you implement appropriate indexes and reorganize any table data and indexes that may suffer from fragmentation.

To examine access methods to tables and indexes

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

2   In the Instance list, choose the item you want to analyze.

3   On the Association controls in the Association area, click `Databases`. Drill down to the database you want to analyze.

4   On the Association controls in the Association area, click `Tables` or `Indexes`. Drill down to a specific table or index entity (depending upon your selection).

5   On the View controls in the Main area, click `Usage Report` to observe information on the access methods used on the table or index.

See [About viewing usage information and access methods in the Usage Report](#) and [About viewing a usage report](#).

Examining the distribution statistics of indexes

Distribution statistics of an index can help you determine whether or not the index is useful to the query optimizer. The information displayed in the Statistics view of the Index entity indicates the selectivity of an index; the more selective an index is, the more useful it is, since this means more rows can be eliminated from consideration.

See [About viewing distribution statistics for a selected index](#).

To examine the distribution statistics of indexes

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

2   In the Instance list, choose the item you want to analyze.

3   On the Association controls in the Association area, click **Databases**. Drill down to the database you want to analyze.

4   On the Association controls in the Association area, click **Indexes**. Drill down to a specific index.

5   On the View controls in the Main area, click **Statistics** and observe the distribution statistics for the selected index in the Distribution table.

Determining whether to recommend an index for a table

The decision to add a specific index to a table is complex. The index can improve one statement and may negatively affect another. Therefore, before adding an index, you should examine every statement accessing the table to see how the new index will impact it.

Precise for SQL Server uses the Microsoft® Index Tuning Wizard to obtain recommended indexes and statistics for a selected table. The recommendations are based on the statements that were executed during the selected time period that have an average duration time greater than the predefined threshold, that refer to the PMDB table. "0" seconds is the default.

By associating to statements from the selected table you can view all the statements using the specified table and their resource consumption. This information can help you estimate which statements will be affected by the new index and how they will be affected.

To determine whether to recommend an index for a table

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

2   In the Instance list, choose the item you want to analyze.

3   On the Association controls in the Association area, click **Databases**. Drill down to the database you want to analyze.

4   On the Association controls in the Association area, click **Tables**. Drill down to the table you want to analyze.

5   On the Recommend tab, analyze the statements that are associated with the selected table and observe their resource consumption.

The Recommend Indexes output shows a list of recommended indexes and statistics. For each recommended item, detailed information is displayed in the Details area, such as a list of key columns and the DDL of the Create Index or Statistics text.

8

Storage Unit entities

This section includes the following topics:

- About Storage Unit entities
- How the Objects tab can help you identify performance problems

About Storage Unit entities

Storage Unit entities can be monitored in the Objects tab, if you have installed Precise for Database & Storage.

If you want to see the Storage Unit entities that are monitored, select Storage Units in the Association area of the Objects tab. The information shown will give the DBA answers on whether there is a contention on a device due to re-tiering by performing the following actions:

Table 8-1 Actions

Action	Indicator	Possible Solution
Examine the device's response time overtime.	Marker on the Response Time graph.	Consider moving heavy files to other devices to split the I/O load or moving the data to a faster storage device.
Examine the device tier distribution and review other devices that share the same policy.	Marker on Tier Distribution graph.	Consider using a policy that secures more space on the high-end storage for the device. Verify if the policy is adequate for the type of activity.
Examine the device tier distribution and review other devices that share the same policy.	Marker on the Association table.	Consider moving other devices that use the fast (EFD) storage to another policy.

Other information related to the Storage Unit entities that can be read from the Objects tab are:

- The type of storage the application is using (Thin/Thick).
- The type of policy that is associated to each device.
- The pools the application is using.
- How full the pools are.
- Who else is using the pools.
- How much of each storage Tier is being used by each device.

About the Association area

In the Association area, you can select the entities from the dropdown menu. When choosing Storage Units to see all of the Storage units (storage arrays) that are connected to the monitored server (either physically or via a remote gatekeeper), two tabs are shown at the right side of the table, Statistics - statistical data about the storage units, and

Configuration - data regarding the unit's configuration. Each tab has its own columns in the table. The following tables describe the different column headings.

Table 8-2 Statistics

Column Name	Description
Type	Type of the device: Symmetrix or Clariion.
Unit ID	The ID of the unit
Used By Instance	Shows whether the Oracle database instance is using the storage unit (the database has files on at least one storage device that belongs to that unit).
Read/Write req.	The number of read and write requests.
Load	The amount of data that the unit handled within the timeframe.
Throughput	The amount of data per second (MB/Sec.).***
I/O Rate	The amount of requests to the unit per second.
Seq. Reads	The amount of sequential requests.
Cache Hit Ratio	The amount of requests that were answered by the cache in relation to the total amount (only for Symmetrix).
Read Hit Ratio	The percentage of read hits in relation to the total cache hit ratio (only for Symmetrix).
Write Hit Ratio	The percentage of write hits in relation to the total cache hit ratio (only for Symmetrix).

Table 8-3 Configuration

Column Name	Description
Type (graphical)	Type of the device: Symmetrix or Clariion.
Unit ID	The ID of the unit
Used By Instance	Shows whether the Oracle database instance is using the storage unit (the database has files on at least one storage device that belongs to that unit).
Unit Physical Disks	The amount of physical disks in the storage array.
Devices	The amount of storage devices.
Type	Type of the device: Symmetrix or Clariion.
Physical Disks	The amount of physical disks***
Devices	The amount of devices***
SQL Server Instances	The amountof SQL Server instances***
Databases	***
Logical Files	***
Model	The unit's model
Microcode Version	The microcode version of the unit.
Cache Size	The size of the cache.
Max. Write Slots	The maximum amount of pending write slots allowed before starting delayed fast writes (only for Symmetrix).***

Table 8-3   Configuration

Column Name	Description
Max. DA Write Slots	The maximum amount of pending write slots allowed for one DA (front director) before starting delayed fast writes (only for Symmetrix).***
Max. Device Write Slots	The maximum amount of pending write slots allowed for one device before starting delayed fast writes (only for Symmetrix).***

The following sections handle the different Storage Device entities:

- About the Storage Unit entity
- About the Symmetrix Unit entity
- About the Symmetrix Device entity
- About the Devices Sharing the Same Policy***
- About the Devices Sharing the Same Pool***
- About the Symmetrix Disk Director entity
- About the Symmetrix Front End Director entity
- About the Symmetrix Remote Director entity
- About the Clariion Unit entity
- About the Clariion Device entity
- About the Pools entity

About the Storage Unit entity

The Storage Unit entity displays a list of all storage entities connected to instances in the Tier. The Storage Unit entity can only be associated to from the Instance level.

About the Storage Unit entity Association area

You can view general details on the entities associated to the selected Storage Units in the Association area. This information is based on the mapping carried out by the Precise for SQL Server Storage agents that are installed.

About the Symmetrix Unit entity

The Symmetrix Unit entity displays general details and I/O activity of the selected Symmetrix unit.

The following views are available:

- Overview
- Performance
- Load
- Throughput

About getting an overview of a Symmetrix unit

The Overview displays general details of the Symmetrix unit, number of I/O and hit ratios for the selected Symmetrix unit, over the selected time period, as collected by the Storage agent, as follows:

Properties	Displays general Symmetrix unit details such as, ID, model, and number of physical disks.
Requests	Displays number of I/O requests of the selected Symmetrix unit, over the selected time period, as collected by the Storage agent.

Hit Ratio   Displays the hit ratio of the selected Symmetrix unit, over the selected time period, as collected by the Storage agent.

About viewing performance information for a Symmetrix unit

The Performance view displays requests counters vs. hit ratio counters of the selected Symmetrix unit broken down into read/writes, over the selected time period, as collected by the Storage agent, as follows:

Requests   Displays number of I/O requests of the selected Symmetrix unit, over the selected time period, as collected by the Storage agent.

Hit Ratio   Displays the hit ratio of the selected Symmetrix unit, over the selected time period, as collected by the Storage agent.

About viewing the I/O load of a Symmetrix unit

The Load view displays I/O load of the selected Symmetrix unit, over the selected time period, broken down into read/written.

About viewing throughput information for a selected Symmetrix unit

The Throughput view displays I/O rate vs. Throughput (in MB) of the selected Symmetrix unit, over the selected time period.

About the Symmetrix Unit entity Association area

You can view general details and statistics counters on the entities associated to the selected Symmetrix Unit entity in the Association area. It is possible to associate to a Symmetrix Unit entity from a Symmetrix Storage Device entity. The following tabs show additional information:

Overview   Displays general details for the selected device.

Statistics   Displays I/O statistics for the selected device as collected by the Storage agent.

About the Symmetrix Device entity

The Symmetrix Device entity displays general details, I/O activity and I/O performance of the selected Symmetrix Storage device.

The following views are available:

- Overview
- Hypers
- Ports
- Performance
- Load
- Throughput
- Scalability

About viewing general details of a selected Symmetrix device

The Overview displays the following information on a Symmetrix device:

Properties   Displays general details of the Symmetrix device such as it's logical name and unit ID.

I/O Wait	Displays the I/O wait data of the selected device, over the selected time period, as collected by the Storage agent.
Requests	Displays number of I/O requests of the selected device, over the selected time period, as collected by the Storage agent.
Hit Ratio	Displays hit ratios of the selected device, over the selected time period, as collected by the Storage agent.
Load	Displays read/write load data for the selected device, over the selected time period, as collected by the Storage agent.

About viewing the list of all hypers included in a Symmetrix device

The Hypers view displays a list of all the Hypers included in the device.

About viewing a list of all the ports included in a device

The Ports view displays a list of all the device ports included in the device.

About viewing the performance of a Symmetrix device

The Performance view displays I/O requests vs. Hit Ratio of the selected device, over the selected time period, broken down into read/writes, as collected by the Storage agent.

About the View load data for a Symmetrix device

The Load view displays read/write data of the selected device, over the selected time period, as collected by the Storage agent.

About viewing throughput data for a Symmetrix device

The Throughput view displays I/O rate vs. Throughput (in MB) of the selected device, over the selected time period, as collected by the Storage agent.

About the scalability of a Symmetrix device

The Scalability view, in the Objects tab (only when choosing a specific device), shows the device's response time in relation to the requests on that device.

About the Symmetrix Device entity Association area

You can view general details and statistics counters on the entities associated to the selected Symmetrix Device entity in the Association area. It is possible to associate to the Symmetrix Device entity from the Database, Table, Index, DB File, File Group, Symmetrix Unit, Symmetrix Device, Symmetrix Disk Director and Symmetrix Front End Director entities. It is possible to associate from the Symmetrix Device entity to the devices on the same Raid Group. The following tabs are available in the Association area:

Overview	Displays general details for the selected device.
Statistics	Displays I/O statistics for the selected device as collected by the Storage agent.

About the Devices Sharing the Same Policy***

This entity is only available, in the Objects tab, if you have installed Precise for Database & Storage. For more information on the supported platforms, see the *Precise Installation Guide*.

About the Devices Sharing the Same Pool***

This entity is only available, in the Objects tab, if you have installed Precise for Database & Storage. For more information on the supported platforms, see the *Precise Installation Guide*.

About the Symmetrix Disk Director entity

The Symmetrix Disk Director entity displays general details and I/O activity of the selected device.

The following views are available:

- Overview
- Performance
- Load
- Throughput

About the overview of a disk director

The Overview, in the Objects tab, displays general details of the disk director, such as director ID and director type.

About the performance of a disk director

The Performance view, in the Objects tab, displays two overtime graphs that illustrate the following information:
- The number of sequential reads and the total number of read and write requests during the selected time period.
- The read, write, and combined cache hit ratios over the selected time period.

About the load of a disk director

The Load view, in the Objects tab, displays an overtime graph of the megabytes of data read and written during the selected time period.

About the throughput of a disk director

The Throughput view, in the Objects tab, displays an overtime graph that illustrates the I/O rate (requests per second) and the throughput (megabytes per second) during the selected time period.

Associating a disk director with other entities

You can associate a disk director in the Main area of the Objects tab with devices. You can display additional entities by using the More... option.

About the Symmetrix Front End Director entity

The Symmetrix Front End Director entity displays general details and the I/O activity of the selected director.

About getting an overview of the Symmetrix Front End Director

The Overview displays general details of the Symmetrix front-end director, along with the number of I/O requests, hit ratio and I/O rate of the selected front-end director, as follows:

Properties	Displays general details of the Symmetrix front-end director, such as Director ID, and Director Type.
Requests	Displays the number of I/O requests for the selected front-end director, over the selected time period, as collected by the Storage agent.
Hit Ratio	Displays the hit ratio of the selected front-end director, over the selected time period, as collected by the Storage agent.

I/O Rate	Displays the I/O rate, of the selected front-end director, over the selected time period, as collected by the Storage agent.

About the Symmetrix Front End Director entity Association area

You can view general details and statistics counters on the entities associated to the selected Symmetrix Front End Director entity in the Association area. It is possible to associate to the Symmetrix Front End Director entity from the Symmetrix Unit and Symmetrix Device entities. The following tabs are displayed:

Overview	Displays general details for the selected device.
Statistics	Displays I/O statistics for the selected device as collected by the Storage agent.

About the Symmetrix Remote Director entity

The Symmetrix Remote Director entity displays general details and the I/O activity of the selected director.

About getting an overview of the Symmetrix Remote Director

The Overview displays general details of the Symmetrix storage remote director, along with the hit ratio, and I/O rate of the selected remote director, as follows:

Properties	Displays general details of the Symmetrix storage remote director such as director ID, and director type.
Requests	Displays the number of I/O requests for the selected remote director, over the selected time period, as collected by the Storage agent.
Hit Ratio	Displays the hit ratio of the selected remote director, over the selected time period, as collected by the Storage agent.
I/O Rate	Displays the I/O rate of the selected remote director, over the selected time period, as collected by the Storage agent.

About the Symmetrix Remote Director entity Association area

You can view general details and statistics counters on the entities associated to the selected Symmetrix Remote Director entity in the Association area. It is possible to associate to the Symmetrix Remote Director entity from the Symmetrix Storage Unit entity. The following tabs are displayed:

Overview	Displays general details for the selected device.
Statistics	Displays I/O statistics for the selected device as collected by the Storage agent.

About the Clariion Unit entity

The Clariion Unit entity displays general details and the I/O activity for the selected unit.

The following views are available:

- Overview
- Performance
- Load
- Throughput

About getting an overview of the Clariion unit

The Overview displays general details of the Clariion unit such as its ID, and the number of SQL Server instances connected to it, as follows:

Requests   Displays the number of I/O requests for the selected Clariion unit, over the selected time period, as collected by the Storage agent.

Load   Displays the I/O load for the selected Clariion unit, over the selected time period.

About viewing the performance of a Clariion unit

The Performance view displays the I/O requests (broken down into read/write operations) for the selected Clariion units, over the selected time period, as collected by the Storage agent.

About viewing the load data of a Clariion unit

The Load view displays the I/O load (broken down into read/write operations) for the selected Clariion unit, over the selected time period.

About viewing throughput data of a Clariion unit

The Throughput view displays the I/O rate vs. throughput (in MB) of the selected unit, over the selected time period, as collected by the Storage agent.

About the Clariion Unit entity Association area

You can view general details and statistics counters on the entities associated to the selected Clariion Unit entity in the Association area. It is possible to associate to the Clariion Unit entity from a Clariion Storage Device entity. The following tabs are displayed:

Overview   Displays general details for the selected device.

Statistics   Displays I/O statistics for the selected device as collected by the Storage agent.

About the Clariion Device entity

The Clariion Device entity displays general details and the I/O activity of the selected device.

The following views are available:

- Overview
- Performance
- Load
- Throughput

About getting an overview of a Clariion device

The Overview displays general details of the Clariion device such as unit ID, and number of DB files on the device, as follows:

I/O Wait   Displays the I/O wait for the selected device, over the selected time period, as collected by the Storage agent.

Requests   Displays the number of I/O requests for the selected device, over the selected time period, as collected by the Storage agent.

Load   Displays the I/O load (broken down into read/write operations) for the selected Clariion device, over the selected time period.

About viewing the performance of a Clariion device

The Performance view displays the I/O requests (broken down into read/write operations), for the selected Clariion device, over the selected time period, as collected by the Storage agent.

About viewing load data of a Clariion device

The Load view displays the I/O load of the selected Clariion device (broken down into read/write operations), over the selected time period.

About viewing throughput data of a Clariion device

The Throughput view displays the I/O rate vs. throughput (in MB) of the selected device, over the selected time period, as collected by the Storage agent.

About the Clariion Device entity Association area

You can view general details and statistics counters on the entities associated to the selected Clariion Device entity in the Association area. It is possible to associate to the Clariion Device entity from the Database, Table, Index, Logical File, File Group and Clariion Storage Unit entities.

About the Devices on the same RAID Group entity (only for Clariion)

This entity is only available, in the Objects tab, if you have installed Precise for Database & Storage. For more information on the supported platforms, see the *Precise Installation Guide*.

When viewing a Thick Clariion Device (in the configuration tab), there is an option called "Devices on the same Raid Group" in the dropdown menu, that shows all of the devices who share the same Raid group as the selected device.

Note: The graphs in the upper part of the screen (Overview, Performance, Load, and Throughput) - are the same graphs of that particular selected device - with no connection to the "Devices on the same Raid Group" option.

About the Pools entity

The Pools entity, in the Objects tab, is only available if you have installed Precise for Database & Storage. For more information on the supported platforms, see the *Precise Installation Guide.*

The following views are available:

- Overview
- Performance
- Load
- Throughput

About the overview of a pool

The Overview, in the Objects tab, displays general details of the pool.

About the performance of a pool

The Performance view, in the Objects tab, displays two overtime graphs that illustrate the following information:

- The number of deferred writes, read requests, and write requests during the selected time period.
- The read, write, and overall cache hit ratios over the selected time period.

About the load of a pool

The Load view, in the Objects tab, displays an overtime graph that illustrates the megabytes of data read and written during the selected time period.

About the throughput of a pool

The Throughput view, in the Objects tab, displays an overtime graph that illustrates the I/O rate (requests per second) and throughput (megabytes per second) during the selected time period.

Associating a pool with other entities

You can associate a pool in the Main area of the Objects tab with all the devices that are on the same disk as the hyper itself. You can display additional entities by using the More... option.

How the Objects tab can help you identify performance problems

The following section describes a few examples on how you can use the Objects tab to identify performance problems.

- [Examining I/O load and performance of the various Storage entities](#)

Examining I/O load and performance of the various Storage entities

In theory, database files should be spread over different physical disks so that the I/O for one database file will not interfere with the I/O of a different database file. Therefore, tuning the I/O on a large storage machine is a complex task. Moving a database file from one disk to another can speed up some parts of the application but may slow down others.

Precise for SQL Server includes storage points that can help you tune the I/O in your system. Storage points are agents that sample activity in the Storage units against which they are installed. They map datafiles to Storage devices and provide data that relates I/O wait of database objects to the Storage devices on which they are stored.

Before moving a database file to another physical disk, you can examine the I/O wait on the database file to find the time range with the most I/O waits. You may also want to examine I/O waits on a physical disk over time to identify a physical disk that experienced either none or very few I/O waits during the selected time period.

The decision as to which file to move and which storage device to move it to is based on many factors, such as free space on the device, performance of the device, load on the device, and other files on the device.

Precise for SQL Server displays all the information you require to reach the optimal decision regarding which storage device to move the file to. The Objects tab maps the storage devices to the logical files stored on them, in addition to the statistics gathered by the Precise for SQL Server Storage agent regarding the various storage devices (such as, Bytes, requests broken down into read/writes, and hit ratio of the storage device).

To examine I/O load and performance of the various Storage entities

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

2   In the Instance list, choose the item you want to analyze.

3   On the Association controls in the Association area, click **Databases**. Drill down to the database you want to analyze.

4   On the Association controls in the Association area, click **Logical Files**. Drill down to a specific logical file.

5   On the View controls in the Main area, click **Overview** and observe the I/O statistics for the selected database file.

Examining SQL Server instance statistics

This section includes the following topics:

About the Statistics tab

The Statistics tab displays statistical information on all SQL Server instances. In addition several OS statistics are also presented to enable you to examine SQL Server and OS statistics together. The tab can be used to monitor your system's current state as well as historical statistical information.

The Statistics tab enables you to provide answers to the following types of questions: "Is SQL Server the major CPU consumer in my server?" or "I am aware that SQL Server suffers from I/O wait. Is it possible to examine the disks' average queue length? "

The statistics tab provides hundreds of raw performance counters (All perfmon counters that are relevant to SQL Server and a few additional OS counters are collected). It is possible to view current status in one-minute intervals or historical information roll-up at a higher summary level.

If you are using SQL Server 2005, you will have access to additional statistics on wait events and latch events. Viewing these statistics will let you diagnose additional performance issues in the In MS-SQL graph. You will be able to view how much time the SQL Server waited for each wait group. Based on the information displayed in the CPU Pressure graph, you will be able to determine if the system is experiencing CPU pressure that may be alleviated by faster CPUs, by adding additional CPUs, or by SQL tuning.

While examining a single counter is very important, the real value obtained from working with the Statistics tab lies in examining the ready-to-use set of graphs that display several related counters.

Use the Statistics tab to periodically monitor the health of your system—for example, view the statistics on CPU utilization or Hit Ratios. Alternatively, you can use the Statistics tab to fully analyze a performance problem reported by the Collector. For example, if the Collector identifies a massive Using CPU State it may be necessary to check the CPU queue length or SQL compilation counters to check related problems that can influence the Using CPU State parameter.

How the Statistics tab is structured

The Statistics tab displays information on a selected entity and its associated entities. When you open the Statistics tab from the Dashboard tab, the tab is launched, in context, and depending upon your selection in the All SQL Server Instances table (All or Instance), the information is displayed on the Tier or Instance level. When you open the Statistics tab from the Activity tab, the tab is launched, in context, if the selected entity exists in the Statistics tab. If you

open the Statistics tab from another tab, the historical settings (meaning those settings which were selected when you left the tab, such as the last entity you drilled down to) are taken into account and the information displayed the last time you viewed this tab is displayed (similar to clicking the History button and returning to a previous tab).

See About the Precise bar.

The selected entity is always reflected in the Tab heading, which serves as a point of orientation. The highest-level entity you can view information for in the Statistics tab is the Tier level (viewed when All Instances is selected from the Instance list). You can also select a specific instance from the Instance list.

The information displayed in the Statistics tab refers to the time period you selected in the Time Frame selector.

The entities displayed in the Association area are related to the selected entity displayed in the Main area.

About the Main area in the Statistics 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, the Instance entity provides information about CPU usage and pressure, memory allocation, I/O activity, network activity, locking, and wait and latch events.

About the Association area in the Statistics 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 all Databases or Disks 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.

The following tabs show additional information when All Instances is selected:

- CPU
- Load
- I/O
- Memory
- Network

If you want to view additional information on an associated entity, drill down to it, by selecting the entity's row. A drill-down 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 view information on a specific disk, choose Disks from the Association controls. The Association area changes to display information on the disks associated with the selected Instance. Note that the Tab heading and the Main area remain unchanged.

In the Association area, click the row of the disk you want to view detailed information for. The Tab heading indicates the newly selected entity; the Main area displays statistical information on the disk entity you drilled down to, and the Association area shows information on its associated performance counters.

See About drilling down in context.

About the entities you can examine

The Statistics tab displays information on the following entities:

- Tier (when All Instances is selected)
- Instance
- Databases
- Disks

- Network Card

- Performance Counter or Counter Instance

- Wait Events

- Latch Events

For an explanation of the counters displayed in the Statistics tab, see the *Microsoft performance monitor (perfmon) tool*

About the Tier entity

Choosing All Instances from the Instances list displays information on the Tier entity.

The Tier entity provides statistical information on all SQL Server instances in the application, in the following views:

- Overview

- Instance Grouping

About viewing a summary of statistical information on top instance consumers

The overview of the Tier entity provides statistical information on the top six instance consumers of various resources and enables you to compare them.

The Tier area shows the top six consumers, as follows:

Top 6 Instances (% CPU)	Displays the top six CPU consuming instances.
Top 6 Instances (Paging)	Displays top six instances with the most page faults.
Top 6 Instances (Batch Requests/sec)	Displays top six instances with the highest batch request rate.

About viewing statistical information on instance grouping

The Instance Grouping view displays performance counters broken down into instance groups (the instance groups are defined by the user). For example, you can view the avg SQL Server CPU usage in the Production group.

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

The following table describes the information displayed in the Instance Grouping table.

Table 9-1   Instance Grouping table

Column	Description
Group	Displays the name of the group.
Instances	Displays the number of instances linked to the group.
SQL Server CPU	Displays the average SQL Server CPU Usage (%) of all the instances in the specified group.
Transactions/sec	Displays the average number of Transactions/sec started for the databases of the instances in the specified group.
Batch Requests/sec	Displays the average number of Batch Requests/sec received by the instances in the specified group
Page Faults/sec	Displays the average number of Page Faults/sec executed by the instances in the specified group.

About viewing instances associated with an Tier in the Association area

You can view information on the Instances associated with the selected Tier in the Association Area.

The following table describes the information displayed in the tabs in the Association area.

Table 9-2 Viewing information on associated instances

Tab	Description
CPU	Displays counters that report on the average and maximum CPU resources used by the SQL Server and the entire Server.
Load	Displays counters that report on instance load such as Login/sec, and Transactions/sec. Both average and maximum values are displayed.
I/O	Displays counters that report on I/O performance, such as the average time it takes to read from the disk, and I/O load, such as the average number of read operations on the disk, per second.
Memory	Displays counters that report on memory utilization by SQL Server, such as SQL Server Memory MB (max) vs. Page Faults per second.
Network	Displays statistics that report on network activity, such as Output Queue Length and Packets sent or received, per second.

See About the Main area and About the Association area.

About the Instance entity

The instance entity displays statistical information for the entire instance. Precise for SQL Server collects hundreds of counters that are relevant to the entire instance. The Instance entity displays predefined graphs that display several counters in each. Each graph contains counters that have some relation to each other.

After drilling down to an instance entity, the following views are available for the Instance entity:

- Overview
- CPU
- Load
- I/O (Performance)
- I/O (Load)
- Memory (Page Faults)
- Memory (Allocation)
- Memory (Buffer Pools)
- Network
- Locking
- Wait Events (SQL Server 2005 only)
- Latch Events (SQL Server 2005 only)
- CPU Pressure (SQL Server 2005 only)

About getting an overview of an Instance based on statistical data

The Overview displays predefined overtime graphs, which can indicate where performance problems lie. The predefined graphs contain counters reported by SQL Server and counters collected by the Precise for SQL Server Collector from perfmon counters. The following graphs are displayed:

In MS-SQL	Displays resource consumption counters collected by the Precise for SQL Server Collector such as, Using CPU, I/O wait, and Lock wait.

CPU	Displays statistics that reports on CPU usage broken down into SQL Server CPU usage and Non SQL Server CPU usage.
Paging	Displays memory paging statistics that are broken down into SQL Server Page Faults/sec and Non SQL Server Page Faults/sec.
Load	Displays instance load statistics such as, Batch Requests/sec, Transactions/sec and Logins/sec.
I/O	Displays I/O load statistics, such as, the percentage of time that has elapsed while the disks are busy servicing read and write requests.
Network	Displays network load statistics, such as Maximum Packets Received/sec and Maximum Packets Sent/sec.

See <u>About session states</u>.

About viewing CPU usage statistics for an Instance

The CPU view displays statistics that report CPU usage. The following graphs are displayed:

CPU	Reports Non-SQL Server CPU time usage and SQL Server CPU time usage.
Processor Queue Length	Displays the Processor Queue Length counter.

About viewing Load statistics for an Instance

The Load view displays statistics that report the instance load. The following graphs are displayed:

Load	Reports Batch Requests/sec, Transactions/sec and Logins/sec.
SQL Compilations	Reports SQL Re-Compilations/sec and SQL Compilations/sec.

About viewing I/O (Performance) statistics for an Instance

The I/O Performance view displays statistics that report on the I/O performance of all the machine's physical disks. The following graphs are available:

Disk sec/IO Operations	Displays Avg. Disk sec/Transfer, Avg Disk sec/Write, and Avg Disk sec/Read.
Disk Queue Length	Displays the Avg. Disk Queue Length counter.

About viewing I/O (Load) statistics for an Instance

The I/O Load view displays statistics that report on Server I/O load and SQL Server I/O load. The following graphs are available:

SQL Server I/O Load	Displays Page Writes/sec and Page Reads/sec.
Disk Operations/sec	Displays Avg Disk Writes/sec and Avg Disk Reads/sec.
I/O	Displays % Disk Write Time and % Disk Read Time.

About viewing Memory (Page Faults) statistics for an Instance

The Memory Page Faults view displays statistics that report instance paging versus memory usage. The following graphs are displayed:

Server Memory	Reports Non SQL Server Memory and SQL Server Memory.
Paging	Reports Non SQL Server Page Faults and SQL Server Page Faults.

About viewing Memory (Allocation) statistics for an Instance

The Memory Allocation view displays statistics that report on memory allocation. The following graphs are available:

SQL Server Memory	Displays SQL Cache Memory, Optimizer Memory, Connection Memory, Lock Memory, Granted Workspace Memory, and Memory in Buffer Pool.
Memory Allocation in Buffer Pool	Free Pages, Stolen Pages, and Database pages.

About viewing Memory (Buffer Pools) statistics for an Instance

The Memory Buffer Pools view displays statistics that report on Buffer Pool allocation and Hit Ratios. The following graphs are available:

Cache and Database Pages	Displays Trigger Plans Cache Pages, Procedure Plans Cache Pages, Prepared SQL Plans Cache Pages, Ad hoc Plans Cache Pages and Database Pages.
Hit Ratios	Displays Trigger Plans Hit Ratio, Procedure Plans Hit Ratio, Prepared SQL Plans Hit Ratio, Ad hoc SQL Plans Hit Ratio, and Buffer Cache Hit Ratio.

About viewing Network statistics for an Instance

The Network view displays statistics that report on network activity. The following graphs are available:

Packets/sec	Displays Max Packets received/sec and Max Packets Sent/sec.
Output Queue Length	Displays the Max Output Queue Length counter.
Bytes/sec	Displays Max Bytes Received/sec and Max Bytes Sent/sec.

About viewing Locking statistics for an Instance

The Locking view displays statistics that report on locks and latches. The following graphs are available:

Latch Waits/sec	Displays the Latch Waits/sec counter. Lock
Timeouts/sec	Displays the Lock Timeouts/sec counter.
Table Lock Escalations	Displays the Table Lock Escalations counter.
Deadlocks and Lock Timeouts	Displays the Deadlocks and Lock Timeouts counter.

About viewing Wait Events for an instance

The Wait Event view displays statistics on the top ten wait events for the selected instance.

About Latch Events for an instance

The Latch Event view displays statistics on the top ten latch events for the selected instance.

About CPU Pressure for an instance

The CPU Pressure view displays statistics on CPU pressure for the selected instance. The stacked graphs show the breakdown of total wait time into the following:

- Wait for Resource
- Wait for CPU
- % Wait for CPU

If the CPU wait time exceeds the resource wait time, consider adding additional CPU or faster CPUs, or perform SQL tuning.

Resource wait time is the amount of time the sessions waited for the resource to become available.

CPU wait time is the amount of time the session waited, from the time the resource became available until the time the session resumed running.

About the Database entity

The Database entity displays SQL Server database statistics. Precise for SQL Server collects many counters that are relevant to the database. The database entity displays predefined graphs that display several counters in each. This enables you to immediately view relevant counters, grouped according to topic.

After drilling down to a database entity, the following views are available:

- Overview

- Transactions

- Log Flushes

- Log Growths or Shrinks

- Replication

- Backup or Restore

About getting an overview of the resource consumption and log activity of a database

The Overview displays overtime graphs that represent resource consumption and log activity of the selected database. The overtime graphs contain counters reported by SQL Server and counters collected by the Precise for SQL Server Collector.

The following information is displayed for the selected database:

General Details	Displays general information regarding the selected database entity.
In MS-SQL	Displays database resource consumption counters collected by the Precise for SQL Server Collector, such as, Using CPU, I/O wait, Lock wait.
Transactions	Displays Transactions/sec and Active Transactions.
Flush Wait Time	Displays the Log Flush Wait Time counter.
Log Flushes	Displays Log Flushes/sec, Log Flush Waits/sec and Log Cache Reads/sec.

See About session states.

About viewing Database Transactions statistics

The Transactions view displays a graph that shows the database's transaction activity, such as Transactions/sec and Active Transactions.

About viewing Database Log Flushes statistics

The Log Flushes view displays statistics that report log performance. The following graphs are available:

Log Flushes	Displays Log Flushes/sec, Log Flush Waits/sec and Log Cache Reads/sec.
Flush Wait Time	Displays the Log Flush Wait Time counter.

About viewing Database Log Growths or Shrinks statistics

The Log Growths or Shrinks view displays statistics that report on the log activity. The following graphs are available:

Log Growths and Shrinks	Displays Log Growths, Log Shrinks and Log Truncations.
Log Bytes Flushed	Displays Log Bytes Flushed/sec counter.

About viewing Database Replication statistics

The Replication view displays statistics that provide information on the replication activity such as number of transactions marked for replication and number of transactions read out from the transaction log and delivered to the distribution database.

About viewing Database Backup or Restore statistics

The Backup or Restore view displays counter that reports on the performance of the backup and restore operations.

About viewing statistical information on the Database entity

You can view information on the entities associated with the selected Database entity in the Association area. For example, you can analyze counters that report on the database's transaction activity and log activity. It is possible to associate to the Database entity from an Instance entity.

See About the Main area and About the Association area.

About the Disks entity

The Disk entity displays predefined graphs that report on disk activity and performance in the Overview view. This enables you to immediately view relevant counters, grouped according to topic.

About getting an overview of Disk performance and I/O load

Displays predefined overtime graphs, which represent I/O load and performance of the selected disk. The following graphs are available:

I/O Wait	Displays I/O wait on the selected disk.
SQL Server I/O Load	Displays Page Read/sec and Page Write/sec.
Disk Queue Length	Displays Avg. Disk Queue Length counter.
Disk sec/IO Operations	Displays Avg Disk sec/Transfer, Avg Disk sec/Read and Avg Disk sec/Write.
I/O Activity	Displays Avg Disk Reads/sec and Avg Disk Writes/sec.
I/O	Displays % Disk Write Time and % Disk Read Time.

About viewing information on entities associated with the selected Disk

You can view information on the entities associated with the selected Disk entity in the Association area. It is possible to associate to the Disk entity an Instance entity. The counters displayed in the Association area report the disk's performance and load.

See About the Main area and About the Association area.

About the Network Cards entity

The Network Card entity displays predefined graphs that report the network's activity, load and latency, in the Overview view. This enables you to immediately view relevant counters, grouped according to topic.

About getting an overview of Disk Network load

The Overview displays statistics that report on the network load. The following graphs are available:

Packets/sec	Displays Packets Received/sec and Packets Sent/sec.
Output Queue Length	Displays the average Output Queue Length counter.
Bytes/sec	Displays Bytes Received/sec and Bytes Sent/sec.

About viewing information on entities associated with the selected Network Card

You can view information on the entities associated with the selected Network Card entity in the Association area. It is possible to associate to the Network Card entity from an Instance entity. The counters displayed in the Association area report the network card's activity, load and latency.

See About the Main area and About the Association area.

About the Performance Counter and Counter Instance entities

The performance counter and counter instance entities display statistical information on the selected counter in the Overview view. Precise for SQL Server collects counters reported by the SQL Server and counters that are relevant to the entire machine. These counters determine which aspects of the objects to monitor. For example, the SQL Server Locks object contains the Lock Requests/sec counter.

Additional counters reported by Precise for SQL Server include:

- Access Methods
- Backup Device
- Buffer Manager
- Buffer Partition
- Cache Manager
- Databases
- Exec Statistics
- General Statistics
- Latches
- Locks
- Memory
- Memory Manager
- Network Interface

- Physical Disk
- Plan Cache
- Process
- Processor
- Replication Agents
- Replication Dist.
- Replication Log Reader
- Replication Merge
- Replication Snapshot
- SQL Statistics
- System
- Transactions
- User Setable

About getting an overview of Performance Counters or Counter Instances

The Overview displays general details of the selected counter and an overtime graph that represents the counter value. The following information is available:

Group	The name of the counter's group.
Average	The average value of the counter, within the selected time frame.
Maximum	Shows the maximum value of the counter, calculated from all time slices in the selected time frame.
	Note: This value may be different from the maximum value shown in the overtime graph to the right because each bar in the graph aggregates a number of time slices.

Minimum	The minimum value of the counter, calculated from all time slices in the selected time frame.
	Note: This value may be different from the minimum value shown in the overtime graph to the right because each bar in the graph aggregates a number of time slices.
Description	The description of the counter.
	Note: If the text in the Explanation field is cut off, hold the cursor over the Explanation description to view all the rest of the text.
Overtime graph	Displays the counter's value over time.

About viewing information on the entities associated with a selected Counter

You can view information on the entities associated with the selected Performance Counter or Counter Instance entities in the Association area. It is possible to associate to the Performance Counter or Counter Instance entities from the Instance, Database, Disk or Network Card entities. When choosing a specific performance group the description of the counters are displayed. When drilling down to a counter with instances, the minimum, maximum and average values for the counter instances are displayed.

See About the Main area and About the Association area.

About the Wait Events entity (SQL Server 2005 only)

The Wait Events entity displays the name of the wait event group, the number of waits experienced by this event group and the average amount of time accumulated by wait events.

About getting an overview of Wait Events

The Overview displays general details of the selected counter. The following information is available:

Wait group	Displays the name of the wait type.
Wait name	Provides a description of what is causing the event.
Times waited (avg)	Average time waited for wait events.
Wait for Resource graph	Displays the time waited for wait events and the time waited for group events.
Waits graph	Displays the number of wait events and the number of wait events for group events.

About the Latch Events entity (SQL Server 2005 only)

The Latch Event entity displays the name of the latch event group, the number of waits experienced by latches in this event group and the average amount of time accumulated by latch waits.

About getting an overview of Latch Events

The Overview displays general details of the selected counter. The following information is available:

Wait group	Displays the name of the latch class.
Wait name	Provides a description of what is causing the event.
Times waited (avg)	Average time waited for latch events.
Wait for Resource graph	Displays the time waited for latch events and the time waited for group events.
Waits graph	Displays the number of latch waits and the number of latch waits for group events.

How the Statistics tab can help you identify performance problems

To determine whether SQL Server is performing optimally it is necessary to analyze performance measurements over time. The Statistics tab provides many performance counters grouped into several predefined graphs that enable you to locate performance problems in your system, such as, CPU, paging, I/O and network statistics. In addition, the In MS-SQL states enable you to correlate between the performance counters and the In MS-SQL states to determine how an activity in the SQL Server instance affects the system. The tab can be used to monitor your system's current state as well as historical state.

Performance tuning examples:

- [Examining memory statistics](#)

- [Examining CPU statistics](#)

- [Examining I/O statistics](#)

- [Examining network activity](#)

- [Examining the hit ratio of the various buffers in SQL Server](#)

- [Examining locking within the SQL server](#)

- [Examining a specific counter over time](#)

- [Examining In MS-SQL wait states](#)

- [Examining In MS-SQL latch wait states](#)

Examining memory statistics

You can examine an SQL Server instance over time to confirm that memory usage falls within the normal range and that no processes, including SQL Server, are lacking or consuming too much memory. Lack of memory or a high memory consumption rate can cause paging. A page fault occurs when a process requires code or data that is not in its working set (its space in physical memory), thereby causing the page to be retrieved from the disk and written to the physical memory. If there is not enough space in the physical memory, a chosen page needs to be written to the disk to free space.

To examine memory statistics

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

2   In the Instance list, choose the instance you want to analyze.

3   On the View controls in the Main area, click **Overview**.
The Page Faults/sec graph displayed in the Overview of the Instance entity shows the page faults broken down into SQL Server page faults and non-SQL Server page faults. Analyze this graph together with the In MS-SQL and Load graphs to determine whether the paging is the source of the performance problems. If you conclude that excessive paging is the source of the performance problems, examine the amount of memory allocated to the SQL Server and reduce it, if necessary.

4   On the View controls in the Main area, click **Memory (Page Faults).**
The Memory (Page Faults) view displays page faults vs. memory allocation broken down into SQL Server and non-SQL Server components. Compare these graphs to determine whether server memory is being allocated properly and whether the physical memory needs to be expanded or reduced to enhance performance.

5   On the Association controls in the Association area click **Memory - Counters.**

6   Drill down to the Available MBytes and Pages/sec counters to analyze memory usage.

Examining CPU statistics

You can examine an SQL Server instance over time to confirm that CPU usage falls within normal range. High CPU usage may indicate the need to improve the application or to add more processors to the machine to enhance performance. A low CPU usage indicates that the system is not using all the machine's processing power.

To examine CPU statistics

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

2   In the Instance list, choose the instance you want to analyze.

3   On the View controls in the Main area, click **Overview.**

The CPU graph displayed in the Overview of the Instance entity shows CPU usage broken down into SQL Server processes and non-SQL Server processes. Analyze this graph together with the In MS-SQL and Load graphs to determine whether CPU usage is the source of the performance problems. If you conclude that high CPU usage is the source of the performance problems, consider adding more CPU or tuning the application in the SmarTune or Activity tabs.

4   On the View controls in the Main area, click **CPU.**

The CPU view displays CPU usage broken down into SQL Server processes and non-SQL Server processes vs. processor queue length. Compare these graphs to determine whether a high level of CPU utilization is causing processor bottlenecks and if the CPU is being consumed by SQL processes or other processes.

5   On the Association controls in the Association area click one of the following:

Process - Counters	To analyze the I/O activity of the SQL Server process.
Processor - Counters	To check CPU balance.
System - Counters	To check queue length.

Examining I/O statistics

You can examine SQL Server I/O activity along with the entire server's I/O activity and disk performance, over time, to detect bottlenecks within the disk subsystem. High I/O activity and poor I/O performance may indicate the need to distribute the files more effectively over the disks to balance the I/O between disks or add more disks to enhance performance.

To examine I/O statistics

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

2   In the Instance list, choose the instance you want to analyze.

3   On the View controls in the Main area, click **Overview.**

The I/O graph displayed in the Overview of the Instance entity displays the I/O activity of the entire machine. Analyze this graph together with the In MS-SQL and Load graphs to determine whether I/O activity is the source of the performance problems. If you conclude that increased I/O activity is the source of the performance problems, examine I/O wait in the SmarTune or Activity tabs, or check if the relevant storage Point is installed, in the Objects tab.

4   On the View controls in the Main area, click **I/O (Load)**.

The I/O (Load) view displays SQL Server I/O activity vs. the entire server's activity. Compare these graphs to determine whether most of the I/O activity is being generated by SQL Server.

5   On the View controls in the Main area, click **I/O (Performance).**

The I/O (Performance) view displays disk sec/IO operations vs. the disk queue length. Compare these graphs to determine whether poor performance is due to a long I/O request queue.

6   On the Association controls in the Association area, click **Disks**. Drill down on a Disk entity to determine if it is experiencing poor performance and high I/O activity and if the load is balanced.

Examining network activity

You can examine the network traffic of the server's network cards, over a specified time period, to detect bottlenecks.

Network performance can be improved by one or more of the following:

- Enhancing the application

- Reducing the amount of data returned

- Configuring the SQL Server instance "network packet size (B)" parameter with a reasonable number that would avoid many packets from being sent or received (causing overhead)

- Reconfiguring the network configuration

To examine network activity

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

2 In the Instance list, choose the instance you want to analyze.

3 On the View controls in the Main area, click **Overview**.

The Network graph displayed in the Overview of the Instance entity shows the maximum packets (of all network cards defined on the server) that were sent or received, during the specified time period. Compare this graph with the In MS-SQL and Load graphs to determine whether the network activity caused the SQL Server's performance to suffer.

4 On the View controls in the Main area, click **Network**.

The Network view displays network traffic vs. output queue length. Compare these graphs to determine whether a long queue length is the cause of heavy network traffic.

5 On the Association controls in the Association area, click **Network Cards**. Drill down on a Network Card entity to locate the network cards with the most activity. Check if you can balance activity load across network cards. If not, consider adding a new network card.

Examining the hit ratio of the various buffers in SQL server

You can examine how the memory in the cache and buffer pool is divided between various objects such as procedure plans, ad hoc SQL plans, and trigger plans, by examining allocation pages vs. hit ratio. High memory consumption vs. a low hit ratio may indicate bottlenecks due to a lack of available physical memory for storing frequently accessed pages.

To examine the hit ratio of the various buffers in SQL server

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

2 In the Instance list, choose the instance you want to analyze.

3 On the View controls in the Main area, click **Memory (Buffer Pools).**

The Buffer Pool view in the Instance entity displays cache and database pages to hit ratios.

Examining locking within the SQL server

Locks are held on SQL Server resources, such as rows read or modified during a transaction, to prevent concurrent updating of resources by multiple transactions. To minimize the locking cost, SQL Server locks resources automatically at a level relevant to the transaction. Locking at a lower level of granularity, such as rows, increases concurrency, but has a higher overhead because more locks are needed if many rows are locked. Locking at a higher granularity, such as tables, has an adverse effect on concurrency because locking an entire table restricts access to the entire table by other transactions, but has a lower overhead because fewer locks are being maintained. The Locking view of the Instance entity displays application locks, internal locks, lock escalations and the number of deadlocks to help you understand the locking behavior of your system. For example, if you observe that a large number of locks have timed out and that there are a large number of lock escalations, you can assume that the escalation causes many locks to time out and should therefore be prevented. In addition you can see how internal locks affect the application's locks.

To examine locking within the SQL server

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

2   In the Instance list, choose the instance you want to analyze.

3   On the View controls in the Main area, click **Locking**.
    The Locking view lets you compare locking data to lock escalations.

4   On the Association controls in the Association area, click **Locks - Counter**. View the counters associated with the locks and compare the locks being implemented by the various resources (such as, Table, Key and RID).

5   Drill down to a specific counter to view additional information on that counter, for the specified time period.

6   If you identify a blocking situation, drill down to the Activity tab and continue to explore the Locks.

Examining a specific counter over time

A wide range of performance counters generated by the operating system are collected by the Precise for SQL Server Collector—some belonging to SQL Server and some to the OS itself. These performance counters are sampled every minute and displayed over time in this tab. When loaded into the PMDB, the counters are aggregated and maintained at a 15-minute time slice.

A performance group is an SQL Server performance group resource or an OS performance group, such as an SQL Server Lock. Each performance group contains a few counters that determine which performance aspects to monitor. For example, the SQL Server Access Methods performance group contains counters called "Pages Allocated" and "Pages Split."

Some performance counters have separate instances. An instance refers to a particular occurrence of a counter. For example, the SQL Server Locks performance group has counters such as "Lock Requests/sec" and "Lock Waits/sec," but because several lock levels exist, such as Database, Extent, Page, etc., each counter is divided into instances (in this case, the instances are lock levels).

To monitor a specific counter, choose the required performance group to associate it with, from the Associations list, and then choose the required counter.

To examine a specific counter over time

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

2   In the Instance list, choose the instance you want to analyze.

3   On the Association controls in the Association area, choose the required performance group the counter is associated with and drill down to it to view additional information on that counter, for the specified time period. For example, if you want to view information on the amount of physical memory available to processes running on the computer, on the Association controls, click **Memory - Counters,** and drill down to the Available MBytes counter.

Examining In MS-SQL wait states

You can observe a breakdown of In MS-SQL waits into specific wait events, and examine the top ten wait events in the Wait Event view. You can also compare all the wait events associated with an instance according to the amount of time they waited, the number of times they waited, and the wait group they belong to. This allows you to easily identify the specific wait events the instance is waiting for. You can then tune these wait events, and free resources to the entire instance.

To examine In MS-SQL wait states

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

2   In the Instance list, choose the instance you want to analyze.

3   On the Association controls in the Association area, click **Wait Events**. View the wait events listed in the Association area. Observe to which wait group an event belongs.
    Compare the event in one wait group to an event in a different wait group to observe how much time each wait event spent in the SQL Server.

4   Drill down on a wait event entity and observe its performance over time.

Examining In MS-SQL latch wait states

You can observe a breakdown of In MS-SQL latch wait states into specific latch wait events, and examine the top ten latch wait events, in the Latch Event view. You can also compare all of the latch wait events associated with an instance according to the amount of time they waited, the number of times they waited, and the latch group they belong to. This allows you to easily identify the specific latch wait events the instance is waiting for. If latch wait activity is higher than you expected, consider increasing the amount of memory allocated to the SQL Server.

To examine In MS-SQL latch wait states

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

2   In the Instance list, choose the instance you want to analyze.

   3   On the Association controls in the Association area, click **Latch Events.** View the latch events listed
   in the Association area. Observe to which latch group an event belongs.
   Compare the event in one latch group to an event in a different latch group to observe how much time each wait
   event spent in the SQL Server.

4   Drill down on a latch event entity and observe its performance over time.

5   You can continue your analysis by viewing the hit ratio in the Memory (Buffer Pools) view. If the hit ratio is below
     99% consider adding RAM to the SQL Server. If the hit ratio exceeds 99%, consider improving the speed of
     your I/O system.

Evaluating explain plans

This section includes the following topics:

- About the SQL tab
- How the SQL tab is structured
- About the View tabs
- About tuning actions
- How the SQL tab can help you identify performance problems

About the SQL tab

The SQL tab enables you to evaluate SQL statements and manage a database warehouse of your application's SQL statements together with their respective explain plans.

The process of explaining statements is a prerequisite for tuning. The explain process is designed to clarify the access path chosen for a statement and translate it into a visual medium.

After a statement is explained, explain results are stored in the PMDB. This information includes the objects referenced by the statement and the operations performed on these objects. The statements are automatically explained every day. In addition, the top 30 resource-consuming statements, and the real execution plans, are automatically explained every 15 minutes.

Understanding the execution plan chosen by the SQL Server optimizer is extremely important when tuning your application. You can ensure optimal system performance by ensuring that the best plans are used for your queries.

Precise for SQL Server provides you with a special tab just for this purpose—the SQL tab. It provides several views of the same plan, making the process of analyzing long and complicated queries much easier to do.

To analyze the plan, Precise for SQL Server provides you with a full picture of the objects (table, indexes) participating in the plan. You can change a plan by modifying your query or changing the schema by, for example, the addition of an index. Precise for SQL Server provides you with index recommendations for your statement.

View the historical information of statements showing performance degradation to identify the source of the problem. A change in the schema, volume or execution plan may explain the impact on performance.

Note: The information described for this tab applies equally to statements and batches.

The following table shows from which tabs and entities it is possible to launch to the SQL tab, in context.

Table 10-1   Launch in-context to following entities

Tab	Entities
Current	Session, Statement or Batch.
Activity	Statement, Batch or Plan.
Objects	Statement, Batch, Stored Procedure, Function, or Triggers.

Table 10-1   Launch in-context to following entities

Tab	Entities
SQL	Statement entered manually by the user through "New" action.
SmarTune	The following findings in the Statement Findings table launches to the SQL tab, in context: ■ Heavy operators launch to SQL tab ■ Missing indexes launch to SQL Recommend view ■ Missing statistics launch to SQL Recommend view ■ Table growth launches to SQL History view, with a time frame of two months ■ Table Schema Change May Increase Its Accessing Time launches to SQL History view, with a time frame of two months ■ Increased launches to SQL History view, with a time frame of two months

The following figure shows the error message that is displayed if the SQL tab is opened with no statement or batch, in-context.

Figure 10-1   No statement in context error message

See the following topics for more information:

- [About the Dashboard tab](#)
- [About the Current tab](#)
- [About the Activity tab](#)
- [About the Objects tab](#)

How the SQL tab is structured

The SQL tab enables you to analyze execution plans and explain results so that you can tune statements and achieve optimal results.

The following tabs control the information displayed in the SQL tab:

- View tabs
- Actions menu

About the View tabs

The SQL tab displays different information regarding the selected statement or batch, in different views. A statement is selected when you drill down to it in another tab (from Current, Activity, or Objects tabs) or when you open a new or

existing statement in the SQL tab. Clicking on a view tab displays additional information regarding the statement or batch. Each view has a different layout.

The following information is displayed in each View tab:

Plan	The Plan view lets you display the estimated (for SQL Server 2000 and SQL Server 2005) or actual (for SQL Server 2005 only) execution plan of a statement and various related information such as statistics, referenced objects and operations performed in the execution plan. In addition, performance findings are also displayed.
Recommend	Precise for SQL Server uses the Microsoft® Index Tuning Wizard or Database Tuning Advisor (DTA) to obtain recommended indexes or statistics for the selected statement or batch so that the optimizer will choose a better access plan and improve the performance of the statement or batch. The Recommend Indexes process only makes recommendations with regards to the addition of indexes or statistics. The What-If tab shows the number of statements whose cost would increase and the number of statements whose cost would decrease if the recommendation was implemented. It also shows the cost of implementing the recommendation so that you can weigh the pros and cons of implementing the change.
History	The History view displays resource consumption over time vs. changes in various parameters to determine which changes led to performance problems.
More ...	The Statements view breaks down the access plan of the entire batch into statements and correlates the statements in the access plan with the statements that were captured by the Precise for SQL Server Collector. The statement id of those statements that were captured are displayed in Precise. It is possible to view the access plan of a different statement by clicking on its respective Plan icon.
	The Compare view enables you to compare access plans of a specified statement. By default, the last access plan is displayed in the left pane. The access plan you want to compare it to is displayed in the right pane. The statement's text is displayed at the bottom of each access plan pane and the selected operator is highlighted.

About viewing the execution plan of a statement

The Plan view lets you display the execution plan of a statement and various related information such as statistics, referenced objects and operations performed in the execution plan. It also displays a list of findings for the specified statement or batch.

If you are running SQL Server 2005, the explain process uses actual plans (when available), instead of estimated plans, for batches that were executed against monitored instances.

The Plan view is divided into two panes. The Execution Plan tree is displayed in the left pane. The Details area is displayed in the right pane.

The information displayed in the Details area is controlled by the following information tabs located above the Details area:

- Highlights
- Objects
- Statistics
- More... (includes Operations and Properties Information tabs)

About the execution plan tree

The Plan tree displays the execution plan tree of the specified statement. When you analyze the access plan of a selected statement, you can examine the access path that was chosen by the SQL Server Optimizer. The optimized text that was generated by the SQL Server is displayed instead of the original text of the statement. The Explain result is displayed as an explain tree. Each branch in the explain tree represents one operation.

If you are running SQL Server 2005, the explain process uses XML showplan output to generate the execution plan tree.

Each operation is displayed in the following format:

[<execution order>] <operation type> (<options>) <accessed object> [<cost in percentage>]

A red (critical) icon will appear for a problematic operation. The color of the icon will change depending on the estimated operation costs and predefined thresholds.

Operations are evaluated as follows:

- Statements with total estimated costs greater than '1' are checked for critical operations.

- Statement operations whose estimated cost percentage was greater than 20% of the entire statement are then labeled with a red icon.

It is also possible to display the Execution plan's formatted text, at the bottom of the Execution Plan Tree. The text that relates to the selected step in the plan tree is highlighted. This allows you to view the text of the statement, the execution plan and additional information, such as the objects referenced by the statement, all at once.

About actions that can be performed on the tree

The following actions can be performed on the execution plan tree:

- The Playback controls, located at the top of the tree enable you to freely move within the execution plan of an explained statement.

- By moving the pointer over the execution plan steps you can view a ToolTip that contains statistical information, such as, Estimated Cost and Average Row Size of the specified step.

- Selecting a specific step will highlight its sons and affect the information displayed in the information tabs.

- Clicking on the plus sign (+) located at the bottom of the tree, displays and highlights the statement's formatted text.

About available operation options

The options available for an operation are determined by operator type, operator arguments, and operator predicates.

Precise for SQL Server enables you to obtain a reasonable understanding of the execution plan selected by the SQL Server Optimizer by displaying the operation's options on the execution plan tree.

The following options are available for different operations:

- Clustered Index—The leaf pages of the index hold the actual data, instead of RIDs (Records ID). A table without a clustered index is called a Heap.

- Non-clustered Index—The index consists only of the columns that combine the key and a pointer to the records in the table data. If the table has a clustered index, then the pointer is the key of the clustered index. If the table does not have clustered index, then the pointer is an RID.

- Seeking—The index tree can be used for quickly locating the matching records. Seeking can be done only on indexes.

- Scanning—The leaf pages of the index or the data pages of the table are consecutively scanned.

Note: Seeking is the recommended way to use indexes.

The following table describes the available operation options.

Table 10-2   Operation options

Operation	Description
Sort	The Sort operator presents the sorting of the rows returned from the previously executed operation (appears just below this operator) in a specific order, optionally eliminating duplicate entries.
	Sorting is required when no index that satisfies the requested ordering exists, or when an index scan is more expensive than sorting. It is usually used as the final step to retrieve the fetched data or a prior step for the join or stream aggregate (group by) operators.
	In some cases, the sort operation requires more than the available memory, in which case a temporary table is used. This involves disk I/O, which impacts performance.
	Sorting a large number of rows makes heavy use of the CPU, which can also affect overall server performance.
	The amount of memory needed for sorting can be estimated by multiplying the number of rows by the average row size. Therefore, minimizing the row size by selecting only the necessary fields will decrease the memory required for the sort operation.
Hash joins	A hash join algorithm is used to access a record directly, by creating a hash function. A hash function uses a key as input and returns a hash value, which is then used as a pointer to locate the row directly. There may be different keys with the same hash value. A good hash function almost never returns the same hash value for different keys.
	Hash join algorithms are used as follows:
	■ Joins—Create a hash table for the rows from the first table. For each row from the second table, probe the hash table according the hash function and return rows dictated by the join type.
	■ Remove duplicates—Create a hash table for the rows of the table. Scan the hash table and return only one item for each entry.
	■ Unions—Create a hash table for the rows of the first table. Return rows from the second table that do not exist in the hash table. The second table must not contain duplicate entries.
Nested joins	Nested joins (also called nested loops) use the nested algorithm to join two tables. The nested algorithm works as follows: For each row in the outer table (the first table, appearing directly below the nested loop operator in the explain tree), SQL Server executes the access plan for the inner table (the second one).
	A large number of iterations may indicate a problem in choosing the access plan due to out-of-date statistics.
Merge joins	A merge join algorithm is used to join two tables by merging them. Merge join algorithms read both tables only once; therefore it is required that the input be sorted according to the join predicates.

About viewing the text and performance findings for a statement

The Highlights tab displays the Statement's formatted text and any performance findings identified for the selected statement. The text that relates to the selected step in the plan tree is highlighted. The highlighted color is based on the selected operator type. The formatted text of the selected statement is displayed in the right pane.

When a table is highlighted, all the table's columns, appearing in the text, are highlighted in the same color as the table. The following is an explanation of the major operator types.

About Operators that access a table or index

The table and its columns are highlighted and color-coded. The columns that accessed the table or index are underlined.

About sort operators

All tables whose columns were used by sort operators are highlighted and color-coded. Each table and its corresponding columns have their own color. The columns that participate in the sort are underlined.

About join operators

The methodology for highlighting nested loop operators differs slightly from that of the merge join and hash join. Nested loop operators are highlighted as follows—all tables and their corresponding columns participating in a sub-tree of the outer table (the first operator that is a direct descendant of the nested loop operator) are highlighted in blue; all tables and their corresponding columns participating in a sub-tree of the inner table (the second operator that is a direct descendant of the nested loop operator) are highlighted in red.

The highlighting methodology for the other join operators (hash join and merge join) differs slightly. The first table and all its corresponding columns are highlighted in blue. The columns that used for the join are underlined. The second table and all its corresponding columns are highlighted in red. The columns that were used for the join are underlined.

About TSQL operators

The operators that represent TSQL statements highlight the entire statement they represent.

About viewing which objects are referenced by the execution plan

The Objects tab displays three tables (Tables used in Plan, Indexes of Table, Columns of Table) that list all referenced objects in the execution plan, including their indexes and columns. Statistical details and general details are displayed for each object and its sub-entities.

About tables used in plan

Displays a list of all the referenced tables in the tree.

The following table describes the information displayed for the referenced tables.

Table 10-3   Tables in use

Column	Description
▤	Click to launch to the Objects tab with the selected table in context.
🔍	Click the Locator icon to locate and highlight all the operators in the execution plan that access the specified table.
Used	Indicates whether the specified table is used in the selected operator in the execution plan tree.
!	The Potential Problem icon indicates there is a potential problem in the specified table. A potential problem can be one of the following: ■ Operators of type table scan, index scan, clustered index scan and filter, where the estimated cost is greater than a predefined value (this value can be changed). ■ Operators with warnings.
Table	Displays the full table name in the following format: *([Database Name].[Owner].[Table Name])*.
Cost	Total cost of the access of the steps to the specified table.
Cost (%)	Total cost in percentage of the access of the steps to the specified table.
Rows	Number of rows in the specified table.
Pages Allocated	The number of pages allocated in the disk for the table records, index records, and text or image data. The value might be incorrect due to out-of-date space usage information. If you suspect the value to be incorrect, use the DBCC UPDATEUSAGE command to recalculate the space usage fields. See *SQL Server Books Online for DBCC UPDATEUSAGE,* for additional information.
Pages Used	The number of pages used in the disk for the table records, index records, and text or image data. The value might be incorrect due to out-of-date space usage information. If you suspect the value to be incorrect, use the DBCC UPDATEUSAGE command to recalculate the space usage fields. See *SQL Server Books Online for DBCC UPDATEUSAGE,* for additional information.
Have Local Predicates	Indicates whether the specified table has local predicates in the execution plan.

About indexes defined on a selected table

Displays a list of all the indexes of the selected table displayed in Tables in use.

The following table describes the information displayed in the Indexes Defined On table.

Table 10-4   Indexes defined on

Column	Description
📄	Click to launch to the Objects tab with the selected index in context.
🔍	Click the Locator icon to locate and highlight all the operators in the execution plan that access the specified index.
Used	Indicates whether the specified index is used in the selected operator in the execution plan tree.
Index	Displays the index name.
Pages Allocated	The number of pages allocated in the disk for the index records. The value may be incorrect due to out-of-date space usage information. If you suspect the value to be incorrect, use the DBCC UPDATEUSAGE command to recalculate the space usage fields. See *SQL Server Books Online for DBCC UPDATEUSAGE* for additional information.
Pages Used	The number of pages used in the disk for the index records. The value may be incorrect due to out-of-date space usage information. If you suspect the value to be incorrect, use the DBCC UPDATEUSAGE command to recalculate the space usage fields. See *Books Online for DBCC UPDATEUSAGE* for additional information.
Cost	Total cost of the access of the steps to the specified index.
Cost (%)	Total cost in percentage of the access of the steps to the specified index.
Last Statistics Calculations	The SQL Server collects statistics for each index and updates them automatically. This column displays the date that the statistics for the specified index was last updated.
Index Depth	Number of levels in the index.
Unique	Indicates whether the index is unique.
Clustered	Indicates whether the index is clustered.

About columns in table

Displays a list of all columns in the selected table displayed in Tables in use. When an index is selected in Indexes of Table, the first column constitutes the index column sorted by the position of the column in the index, and marked with an Ascending or Descending icon.

The following table describes the information displayed in Columns in Table.

Table 10-5   Columns table

Column	Description
A↑	Indicates whether the column is part of the selected index and is sorted in ascending order.
D↓	Indicates whether the column is part of the selected index and is sorted in descending order.
Column	Displays the column name.
Type	Displays the physical storage type of the column.
Key Number	Displays the key number of the index column.
Used in Operator	Indicates whether the column is used in the selected operator in the Execution plan tree.
In Clause	Displays a list of all the clauses that the specified column participates in.

About viewing statistical information on all operators in the execution plan

The Statistics tab displays statistical information of all operators in the Execution plan tree. When you sort a table, you can locate operators by their estimated cost or by any other column in the table.

The following table shows the information displayed in the Execution plan statistics table.

Table 10-6 Execution plan statistics

Column	Description
🔍	Locates and highlights the operator in the execution plan tree that matches the selected operator in the grid.
Operator ID and Type	Displays the operator ID and type.
Subtree Cost	Displays the cost of the operators and its sub-operators.
Estimated Cost	Displays the estimated cost of the current operation. A high cost value may indicate a problem in the current implementation of the operation. If the cost value is greater than "1", it is highlighted in red. Check the Estimated I/O cost and Estimated CPU cost values to determine whether the operation is an I/O consuming operation or a CPU consuming operation (or both).
Estimated Cost (%)	Displays the estimated cost (in percentage) of the current operator.
Estimated CPU Cost	Displays the estimated CPU cost of the current operation. A high cost value may indicate a problem in the current implementation of the operation. If the cost value is greater than "1", it is highlighted in red.
Estimated I/O Cost	Displays the estimated I/O cost of the current operation. A high cost value may indicate a problem in the current implementation of the operation. If the cost value is greater than "1", it is highlighted in red.
Estimated Rows	Displays the estimated number of rows returned from the current operation.
Avg Row Size	Displays the estimated average row size of the rows affected by current operation.
Parallel	Indicates whether the operator is running in parallel.

About displaying information on operation types (More tab > Operations)

The Operations option displays statistical information broken down into operation types of all operators in the Execution plan tree.

The following table shows information on major operation types.

Table 10-7 Major operations

Column	Description
🔍	Locates and highlights the operators in the Execution plan tree that match the selected major operation type.
Major Operation Type	Displays the major types of the operators. The operators in the Execution plan tree are separated into the following major types: ■ Table (any access to table) ■ Index (any access to index) ■ Sort (any sort operation) ■ Join (any join operation) ■ Other (all other operations)
Estimated Cost	Displays the total estimated cost breakdown according to major operation types.
Operations	Indicates the number of operators of specified major operation type.
Objects	Indicates the number of referenced tables/indexes of the specified operation type. This column is only relevant for Table and Index operation types.

The following table shows information on regular operation types.

Table 10-8   Operations

Column	Description
🔍	Locates and highlights the operators in the Execution plan tree that match the selected operation type.
Operation Type	Indicates type of operator, such as, Clustered Index Scan, Index Seek, Nested Loop, etc.
Major Operation Type	Displays the major types of the operators. The operators in the Execution plan tree are separated into the following major types: ■ Table (any access to table) ■ Index (any access to index) ■ Sort (any sort operation) ■ Join (any join operation) ■ Other (all other operations)
Estimated Cost	Total estimated cost breakdown according to operation types.
Operations	Indicates the number of operators of the specified operation type.
Objects	Indicates the number of referenced tables/indexes of the specified operation type.

About viewing general properties of a selected statement (More tab > Properties)

The Properties option displays general details of the selected statement.

The following table shows the properties displayed for a selected statement.

Table 10-9   Selected statement properties

Field	Description

Statement or Batch	Displays the following information:
	■ Statement ID
	   Displays the name of the statement or batch that the SQL tab was launched with. Can also be the name of the statement opened manually.
	■ Database
	   Indicates the database that was in use when the statement was run.
	■ Parsing User
	   Indicates the user that was in use when the statement was run.

Cabinet	Indicates the cabinet in which the statement was saved. This field is only displayed for statements that were edited and saved in the statement workshop (using the Open, Save As, or Edit options).
Folder	Indicates the folder in which the statement was saved. This field is only displayed for statements that were edited and saved in the statement workshop (using the Open, Save As, or Edit options).

Access Plan	Displays the following information:
	■ Estimate Cost
	   Indicates the total estimate cost of the statement as calculated for the last access plan.
	■ Most Recent Show Plan
	   Indicates the last time the statement or batch was explained.
	■ Plan Time
	   Indicates when the access plan was saved. An access plan is saved in two cases: 1) there is no access plan for the statement and this is the first time it is saved; 2) changes were made to the statement's access plan.

About viewing recommended indexes or statistics for a statement or batch

Precise for SQL Server uses the Microsoft® Index Tuning Wizard to obtain recommended indexes or statistics for the selected statement or batch so that the optimizer will choose a better access plan and improve the performance of the statement or batch. The Recommend Indexes process only makes recommendations with regards to the addition of indexes or statistics.

The Recommend view is divided into the following areas:

- Recommended Indexes/Statistics area

- Recommendation details area

About the Recommended Indexes/Statistics area

Displays the indexes or statistics suggested by the Recommend process. The following information is displayed in the Recommended Indexes/Statistics area:

What-if	Select this check box to determine which recommendations you want to test before actually implementing them. (SQL Server 2005 only)
Recommendation type icon	Icon indicating what type of recommendation is being considered for the table.
Object	Displays the full name (including owner) of the table, partition, or schema, to which the recommendation refers.
Clustered	Indicates whether the index is clustered. This parameter is only relevant for create index recommendations.
Unique	Indicates whether the index is unique. This parameter is only relevant for create index recommendations.
Details	Displays a list of the index's or statistic's columns, separated by a comma. The sort order (ascending/descending) is also displayed for recommended indexes.
	Displays the name of the index, for the create drop index recommendation.
	Displays the name of the file group that the schema defines, for the create partition schema recommendation.
	Displays the range defining how the schema should be partitioned, in the create partition function recommendation. Indicates whether data falling on the maximum value in the range will be assigned to the next partition or to the same partition.

About the recommendation type icons

The following icons indicate what type of recommendation is being considered for the table:

	Indicates that the recommendation is to create an index.
	Indicates that the recommendation is to create a statistic.
	Indicates that the recommendation is to create a partition function (for SQL Server 2005 only).
	Indicates that the recommendation is to create a partition schema (for SQL Server 2005 only).
	Indicates that the recommendation is to drop an index (for SQL Server 2005 only).

About the Recommendation details area

Provides details on the indexes or statistics suggested by the Recommend process.

The following tabs enable you to view additional details regarding the recommended indexes or statistics:

- Columns

- DDL

- Table's Indexes/Statistics

- Text
- What-If

About the Columns tab

Displays a list of all the columns in the recommended index's/statistic's table with the following information:

Sort order icon	Indicates whether the column is part of the selected index and the order in which it was sorted.
	If you are using SQL Server 2005, it is possible to add no-key columns that are not part of the recommended index. In this case, an include icon is displayed, indicating that the column is part of the INCLUDE statement.
Key Number	Indicates the key number of the column in the index.
Column	Displays the name of the column.
Type	Displays the physical storage type of the column.
Part of Index	Indicates whether the column participates in an existing index.
First Key of Index	Indicates whether the column participates as the first key in an existing index.
Used	Displays the manner in which the column participates in the statement, according to the following types: Range Scan, Table Scan, Sort and Join.

Note: This tab is not available for create partition function or create partition schema recommendations.

About the sort order icons

The sort order icons indicate whether the column is part of the selected index and the order in which it was sorted. If you are using SQL Server 2005, it is possible to add no-key columns that are not part of the recommended index. In this case, an include icon is displayed, indicating that the column is part of the INCLUDE statement.

Table 10-10 Sort order icons

Sort order icon	Description
A↑	Indicates whether the data in the column will be stored in the index in ascending order.
D↓	Indicates whether the data in the column will be stored in the index in descending order.
In🔍	Indicates that column is part of the INCLUDE statement.

About the DDL tab

Displays the DDL text of the recommended index or statistic or the SQL Server 2005 recommendation.

About the Table's Indexes/Statistics tab

Displays a list of the existing indexes or statistics for the table selected in the recommendation area. The following information is displayed:

	Indicates that the item is an index.
	Indicates that the item is a statistic.
Name	Displays the name of the index or statistic.
Clustered	Indicates whether the index is clustered. (This parameter is not relevant for a statistic.)
Unique	Indicates whether the index is unique. (This parameter is not relevant for a statistic.)
Columns	A list of the columns of the index or statistic columns separated by a comma. If the recommended item is an index, the sort order (ascending/descending) is also displayed.

Note: This tab is not available for create partition function or create partition schema recommendations.

About the Text tab

Displays the selected statement's text or batch's text.

Evaluating a set of recommendations using the What-if tab

The What-If tab shows the number of statements whose cost would increase and the number of statements whose cost would decrease if the recommendation was implemented. It also shows the cost of implementing the recommendation so that you can weigh the pros and cons of implementing the change.

Use the What-If option to evaluate a set of recommendations on the workload containing the most common Transact-SQL statements used by the application, during the selected time frame.

To evaluate a set of recommendations using the What-if tab

1   In the Recommendations table in the left pane, select the check boxes of the recommendations you want to test.

2   Click the What-If tab to view a pie chart, indicating the distribution of the cost of all statements that were evaluated. The Analyzed Statements table displays the explained statements that were aggregated by the collector, that access the objects whose recommendations are being evaluated.

3   Continue your analysis of a particular statement on the SQL tab Plan tab, by clicking the launch icon.

To re-evaluate a different set of recommendations

1   In the Recommendations table in the left pane, select the check boxes of the recommendations you want to test. Clear the check boxes of the recommendations you do not want to test.

2   On the What-If tab, click the What-If link.

Note: If you change the time frame, the information displayed in the What-If tab is automatically calculated and the display is refreshed.

About observing the history of resource consumption

The History view displays resource consumption over time vs. changes in various parameters to determine which changes led to performance problems.

The History view is divided into the following areas:

- Main area
- Change history area

About the Main area

The Main area (upper area) displays resource consumption over time vs. cost and changes over time.

About viewing a history of resource consumption over time

The Main area displays resource consumption over time in the following views:

History	Displays resource consumption over time in the following graphs:

- In MS-SQL
  Displays the resource consumption of the selected statement over a specified time period.
- Cost
  Displays the estimated cost of the selected statement over a specified time period.
  Displays the actual and estimated costs for SQL Server 2005 statements.
- Changes vs. Duration (avg)
  Displays change indications for the statement and its related objects vs. average duration over a specified time period.
- Executions
  Displays the number of executions of the selected statement over a specified time period.

Text	Displays the statement's text.

About the Change History area

The Change History area (lower area) displays all the changes made to various parameters over a specified time period.

Information on the following types of changes is displayed:

- All changes
- Schema changes
- Estimated access plan changes
- Actual access plan changes
- Instance parameters changes
- Database options changes
- Volume changes

About viewing all changes

The All Changes view displays a list of all changes made to various parameters over a specified time period. The changes can be schema changes, access plan changes, instance parameters changes, database changes and volume changes.

The following table describes the information displayed in the All Changes table.

Table 10-11   All Changes table

Column	Description
📄	If the object that was changed is one of the entities in the Objects tab (Instance, Database, Table and Index), the Launch icon opens the Objects tab with the selected object, in-context. Also, if the change type is "Access Plan Changed" the icon will launch the Compare View with the specified access plan compared to the last access path, as detected by the "Explain Statements" process.
Date	Indicates the date the change was detected. Click on the link to switch to Plan view and display the first access plan that was created by the Explain Statements process after the specified change was detected.
Change Type	Displays the type of change detected. Can be one of the following values: ■ Schema changes ■ Instance parameters changes and database option changes, as collected by the Collect Schema Changes process ■ Estimated access plan changes, as collected by the Explain Statements process ■ Actual access plan changes, as collected by the Explain Statements process (SQL Server 2005 only) ■ Volume changes, as collected by Collect Space Utilization process

Table 10-11   All Changes table

Column	Description
Object	Indicates the object that was changed, for example, database, table, and index.
Details	Displays information about the change that was made. For example if the "Auto Shrink File" option was changed in the specified database the Details column will contain: "Auto Shrink File (From NO to YES).

About viewing schema changes

The Schema Changes view displays a list of all schema type changes made to various parameters over a specified time period.

The following table describes the information displayed in the Schema Changes table.

Table 10-12   Schema Changes table

Column	Description
📄	If the object that was changed is one of the entities in the Objects tab (Instance, Database, Table and Index), the Launch icon opens the Objects tab with the selected object, in-context. Also, if the change type is "Access Plan Changed" the icon will launch the Compare View with the specified access plan compared to the last access path, as detected by the "Explain Statements" process.
Date	Indicates the date the change was detected. Click on the link to switch to Plan view and display the first access plan that was created by the Explain Statements process after the specified change was detected.
Change Type	Displays the type of schema change detected. Can be one of the following values: ■ File Group ■ No of Columns ■ Delete Trigger Count ■ Insert Trigger Count ■ Update Trigger Count ■ Column Attribute Changed ■ Index Created ■ Index Dropped ■ Index keys Changed ■ Index Cluster Property Changed ■ Index Unique Property Changed ■ Index Pad Property Changed ■ Index Auto Statistics Property Changed ■ Index Row Lock Disallowed Changed ■ Index Page Lock Disallowed Changed ■ Index Fill Factor Changed
Object	Indicates the object that was changed, for example, database, table, and index.
Details	Displays information about the change that was made. For example if the "Auto Shrink File" option was changed in the specified database the Details column will contain: "Auto Shrink File (From NO to YES).

About viewing estimated access plan changes

The Estimated Access Plan Changes view displays a list of all the access plans of the statement as detected by the Explain Statement process.

The following table describes the information displayed in the Estimated Access Plan Changes table.

Table 10-13 Estimated Access Plan Changes table

Column	Description
TUNE	Click on the tune icon to switch to Plan view and display the first access plan that was created by the Explain Statements process after the specified change was detected.
🔍	The Launch and Compare icon opens the access plan in Compare view and enables you to compare the specified access plan with the last access plan, as detected by the "Explain Statements" process.
Date	Indicates the date the specified access plan was detected by the "Explain Statements" process.
Estimated Cost	Indicates the estimated cost of the specified access plan.
Table Scans	Displays the number of table scans performed in the specified access plan.
Index Scans	Displays the number of index scans performed in the specified access plan.
Parallel	Displays the number of parallel operations performed in the specified access plan.
Nested Loops	Displays the number of nested loops performed in the specified access plan.
Merge Joins	Displays the number of merge joins performed in the specified access plan.
Hash Joins	Displays the number of hash joins performed in the specified access plan.
Sorts	Displays the number of sort operations performed in the specified access plan.

About viewing actual access plan changes

The Actual Access Plan Changes view displays a list of the execution plan of a batch that was actually used when the batch was executed. (SQL Server 2005 only).

The following table describes the information displayed in the Actual Access Plan Changes table.

Table 10-14 Actual Access Plan Changes table

Column	Description
TUNE	Click on the Tune icon to switch to Plan view and display the first access plan that was created by the Explain Statements process after the specified change was detected.
🔍	The Launch and Compare icon opens the access plan in Compare view and enables you to compare the specified access plan with the last access plan.
Date	Indicates the date the specified access plan was executed, as taken from system views. Click on the link to switch to Plan view and display the first access plan that was created by the Explain Statements process after the specified change was detected.
Estimated Cost	Indicates the estimated cost of the specified access plan.
Table Scans	Displays the number of table scans performed in the specified access plan.
Index Scans	Displays the number of index scans performed in the specified access plan.
Parallel	Displays the number of parallel operations performed in the specified access plan.
Nested Loops	Displays the number of nested loops performed in the specified access plan.
Merge Joins	Displays the number of merge joins performed in the specified access plan.
Hash Joins	Displays the number of hash joins performed in the specified access plan.
Sorts	Displays the number of sort operations performed in the specified access plan.

About viewing instance parameters changes

The Instance Parameters Changes view displays a list of all the instance parameter changes detected by the Collect Schema Changes process.

The following table describes the information displayed in the Instance Parameters Changes table.

Table 10-15   Instance parameters changes table

Column	Description
📄	The Launch icon opens the Objects tab with the selected instance.
Date	Indicates the date the change was detected. Click on the link to display the plan after the change occurred. Click on the link to switch to Plan view and display the first access plan that was created by the Explain Statements process after the specified change was detected.
Details	Displays additional information on the Instance Parameters change that was made, including the name of the parameter, and old and new values.

The following parameter changes are displayed:

- Maximum recovery interval in minutes
- Number of locks for all users
- Number of open database objects
- Maximum worker threads
- Network packet size
- Memory for index create sorts (KB)
- Priority boost
- Minimum memory per query (KB)
- Query wait(s)
- Set working set size
- Affinity mask
- Cost threshold for parallelism
- Maximum degree of parallelism
- Minimum size of server memory (MB)
- Maximum size of server memory (MB)
- Maximum estimated cost of query allowed to run by query governor
- User mode scheduler uses lightweight pooling

About viewing database options changes

The Database Options Changes view displays a list of all the database changes detected by the Collect Schema Changes process

The following table describes the information displayed in the Database Options Changes table.

Table 10-16   Database options changes

Column	Description
📄	The Launch icon opens the Objects tab with the selected database.
Date	Indicates the date the change was detected. Click on the link to switch to Plan view and display the first access plan that was created by the Explain Statements process after the specified change was detected.
Database	Displays the name of the database in which the changes occurred.

Table 10-16 Database options changes

Column	Description
Details	Provides additional information on the type of database change that was made, including the name of the database option, and the old and new values.

The following database options changes are displayed:

- Recovery Model Changed
- Auto Update Statistics Changed
- Ansi Null Changed
- Auto Shrink File Changed
- Close Cursors On Commit Changed
- Status Changed
- Auto Create Statistics Changed
- Truncate Log Changed
- Compatibility Level Changed

About viewing volume changes

The Volume Changes view displays a list of all tables whose number of rows were increased by 1000, and whose size increased by 10%, and all indexes whose depth was changed.

The following table describes the information displayed in the Volume Changes table.

Table 10-17 Volume changes table

Column	Description
📄	The Launch icon opens the Objects tab with the selected database.
Date	Indicates the date the change was detected. Click on the link to switch to Plan view and display the first access plan that was created by the Explain Statements process after the specified change was detected.
Object	Indicates the object that was changed.
Details	Displays information about the change that was made.

Comparing access plans of a specified statement

The Compare view enables you to compare access plans of a specified statement. By default, the last access plan is displayed in the left pane. The access plan you want to compare it to is displayed in the right pane. The statement's text is displayed at the bottom of each access plan pane and the selected operator is highlighted.

⋯	The Compare icon displays the list of execution plans available for a statement.

To compare access plans of a specified statement

1   Click the Compare icon.

2   In the Compare Statement dialog box, choose the access plan you want to use as a comparison, from the list.

3   Click OK.
      The statement you selected is displayed in the right pane. The statement's text is displayed in the bottom pane.

The following table describes the information displayed for each access plan in the Compare Statements table.

Table 10-18 Compare statements table

Column	Description
Plan Time	Indicates when the specified access plan was performed.
Plan Type	Indicates whether the specified access plan is an estimated or actual access plan.
Estimated Cost	Displays the estimated cost of the specified access plan.
Table Scans	Indicates the number of table scans performed in the specified access plan.
Index Scans	Indicates the number of index scans performed in the specified access plan.
Parallel Access	Indicates the number of parallel operations performed in the specified access plan.
Nested Loops	Indicates the number of nested loops performed in the specified access plan.
Merge Joins	Indicates the number of merge joins performed in the specified access plan.
Hash Joins	Indicates the number of hash joins performed in the specified access plan.
Sorts	Indicates the number of sort operations performed in the specified access plan.

Note: It is also possible to access the Compare View from the Access Plan Changes table. Click the Compare icon in the Access Plan Changes table to open the Compare view with the selected access plan, in-context, and compare it with the last access plan.

About breaking down a batch's access plan into statements

In many cases a statement is part of a long batch (such as sp or adhoc batch). At times you will want to examine a particular statement, and at times you will want to view the statement in the context of the entire batch.

The Statements view breaks down the access plan of the entire batch into statements and correlates the statements in the access plan with the statements that were captured by the Precise for SQL Server Collector. The statement id of those statements that were captured are displayed in Precise. It is possible to view the access plan of a different statement by clicking on its respective Plan icon.

Note: Only statements that were captured by the Precise for SQL Server Collector can be chosen.

When you switch to this view in the context of a batch, all the batch's statements are listed. In this case, all the information displayed is related to the selected batch.

The following table describes the information displayed in the Statements View table.

Table 10-19 Statements view table

Column	Description
	Switches to Plan view, with the selected statement in-context. This icon is only enabled at the batch level and in the statement that was captured by the Collector.
ID	Indicates the ID of the statement in the batch.
Collector Statement	Displays the ID of the statement and batch as assigned by the Precise for SQL Server collector. Click on the link to display the plan after the change occurred.
Physical Operator	Describes how a query or update was executed. The physical operator describes the physical implementation used to process a statement, for example: Batch level, Select, Update, Cond, etc.
Operations	Displays the number of operations in the specified statement.

Table 10-19 Statements view table

Column	Description
Table Scans	Displays the number of Table Scans performed in the specified statement.
Index Scans	Displays the number of Index Scans performed in the specified statement.
Joins	Displays the number of join operations performed in the specified statement.
Sorts	Displays the number of sort operations performed in the specified statement.
Estimated Cost	Displays the estimated cost for the specified statement.

About tuning actions

The Actions menu lets you perform the following operations:

- Create a new statement

- Open an existing statement

- Edit an existing statement

- Re-explain an access plan

- Display the current estimated access plan using Microsoft SQL Server Management Studio (for SQL Server 2005) or Query Analyzer (for SQL Server 2000)

Creating a new statement

You can create a new statement and save it in the PMDB in a logical cabinet and folder hierarchy. You can also rewrite statements and view their access plans.

To create a new statement

1 On the Actions menu, click **New.**

2 In the New Statement dialog box, choose the relevant properties for the new statement from the drop-down lists and enter a statement name, as follows:

Instance	Indicates the name of the instance that the statement belongs to.
Cabinet	Indicates the name of the cabinet that the statement is saved in (creates a cabinet if it does not already exist).
Folder	Indicates the name of the folder that the statement is saved in (creates a folder if it does not already exist).
Name	Indicates the name of the statement (generated randomly).
Database	Indicates the name of the database where the statement is running.
User	Indicates the statement's parsing user.

3 Type the SQL text for the statement in the Text frame.

4 Click **OK**.
The new statement is saved in the PMDB in a logical cabinet and folder hierarchy.

Opening an existing statement

You can view a statement that was saved in the through the New tuning action or through the Save As option in the Current tab.

To open an existing statement

1 On the Actions menu, click **Open**. The Open Statement dialog box is displayed.

2 In the Open Statement dialog box, choose the relevant properties for the statement you want to view from the Instance, Cabinet, Folder and Name drop-down lists, as follows:

Instance	Indicates the name of the instance that the statement belongs to.
Cabinet	Indicates the name of the cabinet that the statement is saved in.
Folder	Indicates the name of the folder that the statement is saved in.
Name	Indicates the name of the statement.
Database	Indicates the name of the database where the statement is running.
User	Indicates the statement's parsing user.

The statement's text is displayed in the Text frame.

3 Click **OK**.

Editing an existing statement

You can edit a statement that was saved in the through the New tuning action or a statement that was collected by the Precise for SQL Server Collector. The statement that was collected by the Collector is saved as a new statement in a logical cabinet and folder hierarchy.

To edit an existing statement

1 On the Actions menu, click **Edit**.

2 In the Edit Statement dialog box, select the statement you want to edit by choosing its relevant properties from the drop-down lists, as follows:

Instance	Indicates the name of the instance that the statement belongs to.
Cabinet	Indicates the name of the cabinet that the statement is saved in.
Folder	Indicates the name of the folder that the statement is saved in.
Database	Indicates the name of the database where the statement is running.
User	Indicates the statement's parsing user.

3 Enter a new statement name in the Name field.

4 Edit the SQL text for the statement in the Text frame.

5 Click **OK**.
The edited statement is saved in a logical cabinet and folder hierarchy.

About re-explaining an access plan

You can generate a new estimated access plan, save it, if it is different from the most recent access plan, and display the most recent access plan.

About displaying the current access plan using Microsoft SQL Server Management Studio or Query Analyzer

You can display the current estimated access plan using Microsoft SQL Server Management Studio (for SQL Server 2005) or Query Analyzer (for SQL Server 2000).

How the SQL tab can help you identify performance problems

After identifying a problematic statement that is slowing down the response time of a specific application, the first step in tuning the statement is to understand the access path that SQL Server chose for the statement. The explain procedure is designed to clarify the access path chosen for a statement and translate it into a visual medium. Therefore you can easily see whether the optimizer chose the proper execution plan. For example, you can see whether the optimizer performed an index seek as expected. In addition, you can see schema changes related to the statement's objects and changes in the instance and database parameters and compare these changes with previous access plans and In MS-SQL over time data for the selected statement to understand how the changes affected a statement's performance. You can also compare previous access plans and locate the operators that have been changed.

Performance tuning examples:

- [Identifying problematic operators in the latest access plan](#)

- [Locating referenced tables that pose potential problems](#)

- [Locating tables with local predicates that are not being used efficiently in the access plan](#)

- [Examining how schema and configuration changes affected statement performance](#)

- [Comparing statement access plans changes that led to performance changes](#)

- [Recommending Indexes or Statistics procedures on a selected statement](#)

- [Locating the most resource-consuming statements in a batch](#)

Identifying problematic operators in the latest access plan

The first step in tuning a statement is to identify problematic operators. Problematic operators are operators whose estimated cost is high or operators that have caused warnings, such as 'index with no statistics', to be issued.

Note: The lower the estimated cost the better. In most cases an estimated cost that is greater than or equal to 1 is considered high and should be improved.

To identify problematic operators in the latest access plan

1   Do one of the following:

- Launch to the SQL tab with a statement in context

- Open the statement you want to analyze in the SQL tab

2   You can try one of the following to improve problematic operators:

- If the operator is a Table Scan, try creating an index.

- If the operator is of Join Type (such as Nested Loop, Merge, or Hash), examine a different joins method.

- If the operator is a Sort operation, check if the sort is required by the application or can be removed. For example, remove distinct or change the union clause to union all.

- If a warning is issued on the operator, try to implement the warning. For example if there is an index with no statistics perform the UPDATE STATISTICS command on the index.

See [About the execution plan tree](#).

Locating referenced tables that pose potential problems

When it is not possible to access the statement, such as in third company products, the only way to improve the statement is by improving access to the data. This can be achieved by performing changes in the referenced objects by examining the operators that access the tables and identifying potential problematic operators.

To locate referenced tables that pose potential problems

1 Do one of the following:

■ Launch to the SQL tab with a statement in context

■ Open the statement you want to analyze in the SQL tab

2 On the Plan tab, click **Objects**.

3 In the Tables Used in Plan, select the Potential Problem icon to highlight problematic operators.

4 Identify the objects that are referenced by the problematic operations and try to tune those objects.

See [About viewing which objects are referenced by the execution plan](#).

Locating tables with local predicates that are not being used efficiently in the access plan

If you are familiar with the application and statements and are confident in your understanding of how the access plan should be executed, you may want to examine the access plan to check if the optimizer accesses the tables in proper order. When joining tables, the fewer the rows that need to be joined, the faster the join operation will be executed. The optimizer selects which table to access first according to the join conditions of the other tables. Making the incorrect decision regarding which table to start with, can lead to a badly tuned statement, since more rows need to be joined.

There are two kinds of predicates: Local and Join.

Local predicates	columns in the `Where` clause are compared to a constant or to a bind variable. (For example: t1.id = 30).
Join predicates	conditions that join tables. A column in table A is compared to a column in table B. (For example: t1.id = t2.id).

To locate tables with local predicates that are not being used efficiently in the access plan

1 Do one of the following:

■ Launch to the SQL tab with a statement in context

■ Open the statement you want to analyze in the SQL tab

2 On the Plan tab, click **Objects**.

3 In the Tables Used in Plan, check if there are any local predicates defined for the table that the optimizer decided to begin the access plan with. If not, and there are local predicates defined for another table, verify that the optimizer begins the access plan with that table, since this means that less rows will be fetched during the process.

See [About available operation options](#).

Examining how schema and configuration changes affected statement performance

Changing schema and configuration parameters can affect the performance of your application. Precise for SQL Server allows you track the schema and configuration changes and compare them with the resource consumption of your statement to understand how the changes related to the statement affected its performance.

To examine how schema and configuration changes affected statement performance

1 Do one of the following:

■ Launch to the SQL tab with a statement in context

■ Open the statement you want to analyze in the SQL tab

2 On the History tab, on the View controls, click **History**.

3 Analyze the statement's resource consumption over time vs. execution cost, schema changes and number of executions over time. This can help you understand what caused the In MS-SQL data to change and to determine whether it was caused by a change that was made or by the frequency with which the statements were executed. If, for example, changes were made to the schema, examine the time the changes were reported and the time the statement's performance began to improve. This can lead you to locating the changes that led to improved statement performance.

4 On the Change History controls, in the Change History area, you can view additional details on the changes made. You can select the launch icon for a particular change and launch to the Objects tab in context, to continue your analysis of the change.

Comparing statement access plans changes that led to performance changes

If you conclude that a change was made to the access plan you can carry your analysis of the change further by selecting the Compare tab.

The Compare view compares two access plans for the same statements or batch. When selecting an operator in one access plan, the corresponding operator in the other access plan is also selected and the statement text is highlighted.

To compare statement access plans changes that led to performance changes

1 Do one of the following:
- Launch to the SQL tab with a statement in context
- Open the statement you want to analyze in the SQL tab

2 On the Compare tab, compare two access plans for the same statements or batch. Select an operator in one access plan. The corresponding operator in the other access plan is also selected and the statement text is highlighted.

Recommending Indexes or Statistics procedures on a selected statement

At times you may want to run the Microsoft Index Tuning Wizard to help you tune a statement. This can be done in the Recommend view.

Precise for SQL Server uses the Microsoft Index Tuning Wizard to obtain recommended indexes and statistics for a selected statement or batch.

To recommend indexes or statistics procedures on a selected statement

1 Do one of the following:
- Launch to the SQL tab with a statement in context
- Open the statement you want to analyze in the SQL tab

2 On the Recommend tab, observe the list of recommended indexes and statistics. For each recommended item, analyze the detailed information displayed in the recommendation details, such as a list of key columns and the DDL's "create index/statistic" text.

Locating the most resource-consuming statements in a batch

The main difference between examining a statement's access plan and a batch's access plan is in the number of operators in the access plan. Understanding an access plan with hundreds of operators is a difficult task. The Statements view can assist you with this task since it shows the access plan statements and their estimated cost. The ID of the statements sampled by the Precise for SQL Server Collector are also displayed.

To locate the most resource-consuming statements in a batch

1 Do one of the following:
- Launch to the SQL tab with a statement in context
- Open the statement you want to analyze in the SQL tab

2 On the Statement tab, observe the access plan statements and their estimated cost.

3 Determine which of the statements have the highest estimated cost and focus your analysis on that statement.

11

Examining SQL Agent activities and other SQL Server activities

This section includes the following topics:

About the Maintenance tab

Almost every SQL Server instance contains user applications along with SQL Server Agent activities (or other activities that the DBA runs to monitor the instance). The Maintenance tab mainly displays information on the SQL Agent activities and other SQL Server activities that are related to the DBA maintenance tasks.

The Maintenance tab enables you to provide answers to the following types of questions: "When was a specific SQL Agent job executed? What was the duration of the last run and which changes in the job led to performance degradation?"

How the Maintenance tab is structured

The Maintenance tab displays information on a selected entity and its associated entities. When you open the Maintenance tab from the Dashboard tab, the selected entity is by default Instance, meaning that information is displayed on the instance level.

When you open the Maintenance tab from the Activity tab, the tab is launched, in context, if the selected entity exists in the Maintenance tab. For example, if you launch the Maintenance tab from the Activity tab after examining a Job's performance, the same job entity will be displayed when you launch the Maintenance tab.

If you open the Maintenance tab from another tab, the historical settings (meaning those settings which were selected when you left the tab, such as the last entity you drilled down to) are taken into account and the information displayed the last time you viewed this tab is displayed (similar to clicking the History button and returning to a previous tab).

The selected entity is always reflected in the Tab heading, which serves as a point of orientation. The highest-level entity you can view information for in the Maintenance tab is Instance. You can select an instance from the Instance list.

The entities displayed in the Association area are associated with the selected entity displayed in the Main area. At times, the relationship between the entity displayed in the Main area and those displayed in the Association area is that of parent to child and sometimes it merely represents that there is a relationship between the selected entity and the entities displayed in the Association area.

About the Main area in the Maintenance tab

The Main area shows comprehensive information on the selected parent entity. You can choose from several views to examine the entity from different angles. You can, for example, focus exclusively on Jobs data (default), or view data on database maintenance plans or alerts.

About the Association area in the Maintenance tab

The Association area provides corresponding information on the child entities. You can view information on one type of child entity at a time, such as jobs or database maintenance plans, by selecting a control from the Association Controls list. The selection you make is reflected in the Association area only; the Main area remains unchanged. For some entities, tabs above the Association area enable you to view additional information: Clicking a tab changes the table columns for the same type of child entities.

From the Association area, you can also drill down to another parent entity by clicking a table row. A drill-down operation affects the whole tab: When you drill down to another entity, the Tab heading reflects the new selection, the Main area displays information on the new parent, and the Association area shows data on the child entities.

For example, when you want to view information on a specific alert, choose Alerts from the Association controls. The Association area changes to display alert-related information. Note that the Tab heading and the Main area remain unchanged. In the Association area, click the row of the alert you want to view detailed information for. The Tab heading indicates the newly selected entity; the Main area displays information on the selected alert, and the Association area shows information, such as jobs executed as a result of the alert.

See [About Precise for SQL Server tabs](#).

About examining Maintenance Tab Entities

The Maintenance tab displays information on different entities. This section provides an overview of all entities, their meaning, and their views.

The following entities can be examined in the Maintenance tab:

- Instance Entity
- Job Entity
- Job Step Entity
- Job Schedule Entity
- Job Run Entity
- Job Step Run Entity
- Operators Entity
- Alert Entity
- Database Maintenance Plan Entity

About the Instance Entity

The Instance entity displays DBA-related information on the monitored SQL Server instance, in the following views:

- Overview
- Job Changes

About getting an overview of the OS and SQL Server instance

The Overview displays general information on the OS and the SQL Server instance. In addition, the status of the Agent (SQL Server Agent, Microsoft search and Distributed transaction coordinator) is reported.

About viewing which Job changes were carried out during a selected time period

The Job view displays the job changes that were carried out during the selected time period. The changes can occur in the job itself, and in the job's steps and schedule details.

The following information is displayed:

Name	Name of the job that was changed.
Change Type	Type of change. The following types of changes are reported—Dropped, Created, Changed. Any change in a job's steps is reported as "Changed", including Create and Drop.
Change Date	The estimated change date. The change is reported by the Precise for SQL Server schema change process, which runs, by default, once a day. In other words, the Change Date is actually the date the schema change process was executed (or the sample date).

About the Job Entity

The Job entity displays information on the selected job. While job performance information is also available in the Activity tab, the Maintenance tab displays additional information, such as, configuration parameters and scheduling, as well as performance information on the job runs. Unlike the Activity tab, where the source of information is the Precise for SQL Server Collector, the information displayed in the Maintenance tab is retrieved directly from the SQL Server.

The following views are available:

- Overview
- Notifications
- Job Changes

About getting an overview of the selected job

The Overview displays information on the selected job along with information on its last run and job runs, over the selected time period, as follows:

Details	Displays the following job details:
	▪ Job—displays the job name.
	▪ Category—displays the category of the selected job. For example, a job can be categorized as "Database Maintenance".
	▪ Enabled—indicates whether the job is enabled or disabled.
	▪ Scheduled—indicates if there is at least one schedule for the job.
	▪ Description—displays the job description entered by the job's creator.
Last Run	Displays information related to the last execution of the job.
	The following information is displayed:
	▪ Date—displays the date of the last run. Last Duration—displays the duration of the last run. Status—displays the status of the last run. Status can be one of the following: Failed, Succeeded, Retry, Canceled, In Progress. Last Step Executed—displays the last step executed during the last execution of the job.
In MS SQL graph	Displays the job's resource consumption over time, within the selected time frame.
Run history graph	Displays the number of job executions versus duration over the selected time period.

About viewing a job configuration to send notifications

A job can be configured to notify various messages when it completes its run. The Notifications view displays how the job has been configured to send its various notification messages.

The following information is displayed:

Operators	E-mail Operator—E-mail address of the operator to notify.
	Page Operator—Computer or user ID used when sending a page.
	Net Send Operator—Computer or user ID used when sending network messages.
	Write to Windows Application Log—Indicates whether notification will be sent to the event log (YES/NO).
	Delete Job—Indicates whether the job should be deleted when it is completed (YES/NO).
Notifications	E-mail Notify—Indicates whether a notification e-mail should be sent, when the job is completed. The following values are available:

- Never—notification will never be sent after the job is completed.
  When the job succeeds.
  When the job fails.
  Whenever the job completes (regardless of the job outcome).

Page Notify—indicates whether a notification page should be sent, when the job is completed. The following values are available:

- Never—notification should never be sent after the job is completed.
  When the job succeeds.
  When the job fails.
  Whenever the job completes (regardless of the job outcome).

Net Send Notify—indicates whether a network message should be sent, when the job is completed. The following values are available:

- Never—notification should never be sent after the job is completed.
  When the job succeeds.
  When the job fails.
  Whenever the job completes (regardless of the job outcome).

Write Notify—indicates whether a notification event should be logged to the Microsoft® Windows NT® application log, when the job is completed. The following values are available:

- Never—the job should never be deleted.
  When the job succeeds.
  When the job fails.
  Whenever the job completes (regardless of the job outcome).

Delete Notify—indicates whether or not the job should be deleted when the job is completed.

About viewing job changes, schedules and step changes

The Job Changes view displays the job changes and job schedules and step changes that were performed over the selected time period. The following information is displayed:

Type	The type of object that was changed—Job, Job step, Job schedule.
Name	Name of the object.
Change Type	Type of change. The following types of changes are reported—Dropped, Created, Changed.
Data Changed	The data that was changed (such as, schedule details, command text.)
Old Value	The value before the change.
New Value	The value following the change.
Change Date	The estimated change date. The change is reported by the Precise for SQL Server schema change process, which runs, by default, once a day. In other words, the Change Date is actually the date the schema change process was executed (or the sample date).

The following job details are monitored by the Collect Schema Changes process:

- Enabled

- Category name

- Owner

- Start step name

- E-mail notify

- Page notify

- Net send notify

- Event log notify

- E-mail operator

- Page operator

- Net send operator

- Delete level

About the Job entity Association area

You can view information on the entities associated with the selected Job entity in the Association area. It is possible to associate to the Job entity from the Instance, DB Maintenance Plan, Operator and Alert entities.

The following table describes the details displayed for each job.

Table 11-1   Job details table

🔧	Indicates whether the job has been changed according to the Precise for SQL Server schema change process, which runs by default once a day
Job	Displays the name of the job. Each row in the table indicates a different job.
Duration	Displays the total time that elapsed for the executions of the job, over the selected time period.
Runs	Displays the number of job executions carried out over the selected time period.
Job Status	Displays the current status of the job (Executing Job Step, Waiting For Thread, Between Retries, Not Running, Suspended).
Owner	Displays the job owner, which can be one of the logins defined in the instance.
Category	Displays the category to which the job belongs, for example, Uncategorized (Local), Database Maintenance, Replication.
Enabled	Indicates whether the job is enabled or disabled.
Scheduled	Indicates whether or not a job is scheduled (YES/NO).
Warnings	Shows the warnings defined for a job. For example, if no schedule was defined for a job the following message is displayed: "No schedules were defined for this job".

The following table shows information on the job's last run.

Table 11-2   Last Run table

Column	Description
Status	Displays an icon indicating the status of the last run. Status can be one of the following: Failed, Succeeded, Retry, Canceled, In Progress.
Job	Displays the job name.

Table 11-2   Last Run table

Column	Description
Last Duration	Duration of the last run.
Date	Date of last run.
Last Step Executed	The last step executed during the last execution of the job.

About the Job Step entity

The Job Step entity displays information on the selected Job step. While Job Step performance information is also available in the Activity tab, the Maintenance tab displays additional information, such as, configuration parameters, what action to perform when the current step has succeeded, in addition to performance information on the job step runs. Unlike the Activity tab, where the source of information is the Precise for SQL Server Collector, the information displayed in the Maintenance tab is retrieved directly from SQL Server.

The following views are available:

- Overview

- Command

- Job Step Changes

About getting an overview of a job step

The Overview displays information on the selected job step along with information on its last run and job step runs, over the selected time period.

Details	Step Name—displays the name of the step.
	Step ID—displays the step ID. Job—displays
	the name of the step's job. Owner—displays
	the owner of the step's job.
	Category—displays the category of the selected job step. For example, a job step can be categorized as "Database Maintenance".
	Type—displays the name of the subsystem used by SQL Server Agent to execute the job step (for example, TSQL, and Replication Distributor.).
	Sample Date—the estimated last change date of one of the job step details. The change is reported by the Precise for SQL Server Schema Change Process, which runs, by default, once a day. In other words, the Sample Date is actually the date the schema change process was executed.
Action	Displays configuration information regarding step execution.
	On Success—action to perform when the execution of the step succeeds. For example, Go to next step.
	On Fail—action to perform when the execution of the step fails. For example, Quit the job when a failure is detected.
	Retry—number of retry attempts to perform if the step fails.
	Retry Intervals—amount of time to wait between retry attempts.
Last run	Displays information on the last execution of the Job Step.
	Date—date of the last run. Duration—
	the duration of the last run.
	Status—displays the status of the last run. Status can be one of the following: Failed, Succeeded, Retry, Canceled, In Progress.
	Message—displays an SQL Server error, if the job step finished with an error, in addition to information on the step run, such as under which user the step was executed. (Note: if the text in the Message field is cut off, hold the cursor over the Message to view the rest of the text.)

In MS-SQL graph   Displays resource consumption for the job step over a specified time frame.

Run history graph   Displays the number of job executions versus duration, over the selected time period.

About viewing the command executed by a subsystem

The Command view displays the command to be executed by the subsystem during the current job step.

About viewing Job Step changes

The Job Step Changes view displays the Job Step changes that were carried out during the specified time frame, as follows:

Data Changed   Displays the data that was changed (such as, command text, and what action to perform in case of failure).

Old Value   The value before the change.

New Value   The value following the change.

Change Date   The estimated change date. The change is reported by the Precise for SQL Server schema change process, which runs, by default, once a day. In other words, the Change Date is actually the date the schema change process was executed (or the sample date).

The following job step details are monitored by the Collect Schema Changes process:

- Sub System Name
- Command Text
- On Success Action
- On Success Next Step
- On Fail Action
- On Fail Next Step
- Database Name
- Database User Name
- Retry Attempts
- Retry Interval

About the Job Step entity Association area

You can view information on the entities associated with the selected Job Step entity in the Association area. It is possible to associate to the Job Step entity from the Job entity.

The following table shows the general job step information displayed in the Details tab.

Table 11-3   General job step details

Column	Description
🔹	Indicates whether the job step has been changed according to the Precise for SQL Server schema change process, which runs by default once a day.
1	Indicates whether the current step is the first step in the job.
Step ID	Displays the ID of the step in the job.
Step Name	Displays the name of the step.

Table 11-3   General job step details

Column	Description
Type	The name of the subsystem used by the SQL Server Agent to execute the job step. For example, TSQL, and Replication Distributor.
On Success	Action to perform when the execution of the step succeeds. For example, Go to next step.
On Fail	Action to perform when the execution of the step fails For example, Quit the job when a failure is detected.
Database	The name of the database in which the command is executed (relevant only if the subsystem is TSQL and Replication Queue Reader).

The following table displays details of the job step's last run.

Table 11-4   Last run tab

Column	Description
	Indicates whether the job step has been changed according to the Precise for SQL Server schema change process, which runs by default once a day.
	Indicates whether the current step is the first step in the job.
Step ID	Displays the ID of the step in the job.
Step Name	Displays the name of the step.
Date	The date and time the job step was last executed.
Status	Status of the job step execution. The status can be one of the following: Failed, Succeeded, Retry, Canceled, In progress.
Message	Displays an SQL Server error, if the job step finished with an error, and information on the step run, such as under which user the step was executed.
Duration	Amount of time that elapsed during the last execution of the job step.

About the Text tab

The following table displays the job step command to be executed by the subsystem.

Table 11-5   Job Step command

Column	Description
	Indicates whether the job step has been changed according to the Precise for SQL Server schema change process, which runs by default once a day.
	Indicates whether the current step is the first step in the job.
Step ID	Displays the ID of the step in the job.
Step Name	Displays the name of the step.
Step Command	Displays the command to be executed by the subsystem during the current job step.

About the Job Schedule entity

The Job Schedule entity displays scheduling information for the selected Job, such as schedule details, and next run date.

The following views are available:

- Overview
- Job Schedule Changes

About getting an overview of a job schedule

The Overview displays the following Job Schedule details:

Job	Displays the name of the job.
Schedule	Displays the name of the job schedule.
Enabled	Indicates whether or not the schedule is enabled (YES/NO).
Creation Date	Displays the date the job schedule was created.
Next Run Date	Displays the date of the next job execution.
Schedule Details	Displays job scheduling details, such as, frequency, date range, time range, and days in week to run the job.
Sample Date	The estimated last change date of one of the job schedule details. The change is reported by the Precise for SQL Server schema change process, which runs, by default, once a date. In other words, the Sample Date is actually the date the schema change process was executed.

About viewing Job schedule changes

The Job Schedule Changes view displays the Job Schedule changes that were carried out during the selected time frame, as follows:

Data Changed	The data that was changed. Old
Value	The value before the change.
New Value	The value following the change.
Change Date	The estimated change date. The change is reported by the Precise for SQL Server schema change process, which runs, by default, once a day. In other words, the Change Date is actually the date the schema change process was executed (or the sample date).

The following job schedule details are monitored by the Collect Schema Changes process:

- Enabled
- Schedule Details

About the Job Schedule entity Association area

You can view information on the entities associated with the selected Job Schedule Changes entity in the Association area. It is possible to associate to the Job Schedule Changes entity from the Job entity.

The following table describes the information displayed on job schedule changes.

Table 11-6   Job Schedule Changes table

Column	Description
	Indicates whether the job schedule has been changed according to the Precise for SQL Server schema change process, which runs, by default, once in a day.
Schedule	Displays the name of the job schedule.
Enabled	Indicates whether or not the schedule is enabled (YES/NO).
Creation Date	Displays the date the job schedule was created.
Next Run Date	Displays the date of the next job execution.

Table 11-6   Job Schedule Changes table

Column	Description
Schedule Details	Displays job scheduling details, such as frequency, date range, time range, and on which days of week job should be run.

About the Job Run entity

The Job Run entity displays information on the selected Job run, such as, run date, duration, last step executed.

About getting an overview of a job run

The Overview displays general information on the Job Run.

Details	Job—displays the job name.
	Owner—displays the job's owner.
	Category—Displays the category of the selected job run. For example, a job run can be categorized as "Database Maintenance".
Run	Date—the date the selected job run was executed.
	Duration—amount of time that has elapsed while the selected job run was executed. Status—displays the status
	of the job run (Failed, Succeeded, Retry, Canceled, In progress). Message—displays general job run
	information, such as whether or not the job was successful, which schedule executed the job run, and the last step that was executed. (Note: if the text in the Message field is cut off, hold the cursor over the Message to view the rest of the text.)
	Last Step Executed—the last step executed in the selected job run.
	Steps Chain—displays the step ID of all the steps executed in the selected job run in the order of their execution time. A plus sign (+) or minus sign (-) follows each step ID indicating whether or not the step was successful. Plus (+) indicates that the step was executed successfully; Minus (-) indicates that the step's execution failed.

About the Job Run entity Association area

You can view information on the entities associated with the selected Job Run entity in the Association area. It is possible to associate to the Job Run entity from the Job entity.

The following table describes information displayed in the Job Run table.

Table 11-7   Job Run table

Column	Description
Date	Displays the date the job run was executed.
Status	Displays an icon indicating the status of the job run (Failed, Succeeded, Retry, Canceled, In progress).
Duration	Amount of time that elapsed during the execution of the job run.
Last Step Executed	The last step executed in the job run.
Steps Succeeded	Number of steps that were executed successfully during the job run.
Steps Failed	Number of steps that failed during the job run.
Message	Displays general information on the job run, such as, whether or not the job was successfully executed, which schedule executed the job run and the last step that was executed. (Note: if the text in the Message field is cut off, hold the cursor over the Message to view the rest of the text.)

About the Job Step Run entity

The Job Step Run entity displays information on the selected Job Step Run, such as, run date, duration of execution, and status.

The following views are available:

- Overview
- Command

About getting an overview of a job step run

The Overview displays general information on the Job Step Run, as follows:

Details	Step Name—displays the step name.
	Step ID—displays the step ID. Job—
	the name of the step's job. Owner—
	the owner of the step's job.
	Category—displays the category of the selected job run. For example, a job run can be categorized as "Database Maintenance".
	Type—the name of the subsystem used by SQL Server Agent to execute the job step. For example: TSQL, and Replication Distributor.
Run	Date—the date the selected job step run was executed.
	Duration—amount of time that elapsed during the execution of the selected job step run.
	Status—status of the job step run (Failed, Succeeded, Retry, Canceled, In progress).
	Message—displays an SQL Server error if the job step run finished with an error, and additional information, such as indicating under which user the step was executed. (Note: if the text in the Message field is cut off, hold the cursor over the Message to view the rest of the text.)

About Command

The Command view displays the command to be executed by the subsystem during the current job step run.

Note: The command displayed is the command currently saved in the production system table. If you change the command after this step is run you will not be observing the command that was actually executed. To view the command that was actually executed, check the Job Step Changes view for the specified step.

About the Job Step Run entity Association area

You can view information on the entities associated with the selected Job Step Run entity in the Association area. It is possible to associate to the Job Step Run entity from the Job Run and Job Step entities.

The following table describes the information displayed in the Job Step Run table.

Table 11-8 Job Step Run table

Column	Description
Date	Displays the date the selected job step run was executed.
Status	Displays an icon that shows the status of the job step run. Status can be one of the following: Failed, Succeeded, Retry, Canceled, In progress
Duration	Displays the time that has elapsed during the execution of the job step run.
Message	Displays an SQL Server error, if the job step run finished with an error, and additional information on the job step run, such as under which user the step was executed.

About the Operators entity

The Operator entity displays information on the selected Operator, such as operator name, E-mail address, and pager address.

About getting an overview of a selected Operator

The Overview displays the following general details on the Operator. The following information is displayed:

Details   Operator—displays the operator's name.

               Enabled—indicates whether or not the operator can receive notifications when an alert occurs (YES/NO).

               E-mail Address—displays the operator's e-mail address.

               Last E-mail Date—displays the date the operator last received an e-mail alert notification.

               Pager Address—displays the operator's pager address.

               Last Pager Date—displays the date the operator last received a pager alert notification.

               Net Send Address—displays the operator's Net Send address.

               Last Net Sent Date—displays the date the operator last received a network message.

About the Operators entity Association area

You can view information on the entities associated with the selected Operators entity in the Association area. It is possible to associate to the Operators entity from the Instance and Alert entities.

The following table describes the information displayed in the Operators table.

Table 11-9   Operators table

Column	Description
Operator	Displays the operator's name.
Enabled	Indicates whether or not the operator can receive notifications when an alert occurs (YES/NO).
E-mail Address	Displays the operator's e-mail address.
Last E-mail Date	Displays the date the operator last received an e-mail alert notification.
Pager Address	Displays the operator's pager address.
Last Pager Date	Displays the date the operator last received a pager alert notification.
Net Send Address	Displays the operator's Net Send address.
Last Net Sent Date	Displays the date the operator last received a network message.

About the Alert Entity

The Alert entity displays information on the selected SQL Server Alert, such as alert name, error number, and E-mail notification.

About getting general information on an SQL Server Alert

The Overview displays general information on the Alert. The following information is displayed:

Details	Alert—displays the name of the alert.
	Enabled— indicates whether the alert is enabled or disabled (YES/NO).
	Error Number—displays the number of the error that occurred.
	Severity—displays the level of severity that triggered this alert.
	Occurrences—number of occurrences for this alert.
	Last Occurred—displays the last occurrence of this alert.
	Count Reset Date—displays the date the occurrence count will be initialized to 0.
	Performance Condition—the condition that is defined in the performance counter that triggers the alert. For example, SQLServer: Access Methods|Extent Deallocations/sec||<|15.
Notification	E-mail Notify—indicates whether or not an E-mail notification will be sent (YES/NO).
	Pager Notify—indicates whether or not a pager notification will be sent (YES/NO).
	Net Send Notify—indicates whether or not a network message will be sent (YES/NO).
	Last Notification—displays the alert's last notification date.
	Notification Message—displays additional information sent with the alert.

About the SQL Server Alert Association area

You can view information on the entities associated with the selected Alert Entity in the Association area. It is possible to associate to the Alert entity the Instance entity.

The following table describes the information displayed in the Details table.

Table 11-10   Details table

Column	Description
Alert	The name of the alert.
Enabled	Indicates whether the alert is enabled or disabled (YES/NO).
Error Number	Displays the number of the error that occurred.
Severity	Displays the level of severity that triggered this alert.
Last Occurred	Displays the last occurrence of this alert.
Occurrences	Number of occurrences for this alert.
Count Reset Date	Displays the date the occurrence count will be initialized to 0.
Performance Condition	The condition that is defined on the performance counters that trigger the alert. For example, SQLServer: Access Methods|Extent Deallocations/sec||<|15.

The following table describes the information displayed in the Notifications table

Table 11-11   Notifications table

Column	Description
Alert	Displays the name of the alert.
E-mail Notify	Indicates whether or not an E-mail notification will be sent (YES/NO).
Pager Notify	Indicates whether or not a pager notification will be sent (YES/NO).
Net Send Notify	Indicates whether or not a network message will be sent (YES/NO).

Table 11-11 Notifications table

Column	Description
Last Notification	Displays the alert's last notification date.
Notification Message	Displays additional information sent with the alert.

About the Database Maintenance Plan entity

The DB Maintenance Plan entity displays information on the selected DB Maintenance Plan such as, owner, databases, and actions.

The following views are available:

- Overview
- History
- Performance

About getting an overview of the database maintenance plan

The Overview displays general information on the maintenance plan. The following information is displayed:

Details	DB Maintenance Plan—the name of the selected DB maintenance plan.
	Creation Date—displays the date the selected DB maintenance plan was created.
	Owner—displays the owner of the selected DB maintenance plan.
	Databases—displays a list of the databases associated to the selected DB maintenance plan.
	Actions—displays a list of the actions to be performed by the selected DB maintenance plan.

About viewing a maintenance plan's history

The History view displays a list of the actions performed by the selected maintenance plan. The following information is displayed:

Action	Activity performed by the selected maintenance plan. For example, Backup database, and Verify Backup.
Start Time	Displays the date and time the action started.
End Time	Displays the date and time the action ended.
Duration	The amount of time that elapsed time during the execution of the action.
Status	Status of the action (Failed, Succeeded).
Message	Displays a message regarding the action. For example, the message displayed for the "Backup database" action is: "Backup Destination [path]". (Note: if the text in the Message field is cut off, hold the cursor over the Message to view the rest of the text.)
Error Number	Displays the error number that is reported if the action failed.

About Performance

The Performance view displays the total duration of the jobs that were executed by the DB maintenance plan, during the selected time period.

See Focusing on information in overtime graphs.

About the Database Maintenance Plan entity Association area

You can view information on the entities associated with the selected DB Maintenance Plan in the Association area. It is possible to associate to the DB Maintenance Plan entity from the Instance entity.

The following table describes the information displayed in the DB Maintenance Plan table.

Table 11-12 DB Maintenance Plan table

Column	Description
DB Maintenance Plan	The name of the selected DB maintenance plan.
Creation Date	Displays the date the selected DB maintenance plan was created.
Owner	Displays the owner of the selected DB maintenance plan.
Databases	Displays a list of the databases associated to the selected DB maintenance plan.
Actions	Displays a list of the actions to be performed by the selected DB maintenance plan.

How the Maintenance tab can help you identify performance problems

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

- Examining job performance
- Examining how job changes affect performance
- Examining the last execution of a job
- Monitoring the occurrences of alerts
- Monitoring a database (DB) maintenance plan

Examining job performance

Examining job runs over time can help you identify problematic time periods and view the general trend of the job runs. Examining how the job ran, comparing its number of executions and total duration, and analyzing a breakdown of its performance in the SQL Server can help you detect bottlenecks and discover their causes. Bottlenecks can be caused by a variety of causes, such as changes in scheduling, running the job at an inappropriate time, and offending SQLs.

To examine job performance

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

2   In the Instance list, choose the item you want to analyze.

3   On the Association controls, in the Association area, click **Jobs**.

4   On the Details tab, drill down to the job you want to analyze.

5   On the View controls in the Main area, click **Overview**.

6   In the Main area, observe how the job ran and compare the number of times it ran with the amount of time it ran. Compare this information to the information displayed the In MS-SQL overtime graph to see a breakdown of its performance in the SQL Server.

7   Locate the run with the highest duration and drill down to it to view additional information.

Examining how job changes affect performance

Changing a job's details may affect the performance of your system. Some changes are implemented to improve the performance of your application. Precise for SQL Server allows you track job changes and correlate them to the resource consumption of the job. This allows you to detect changes that increase the resource consumption of the job.

The jobs that were changed are indicated by an icon in the Association area.

To determine how job changes affect performance

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

2 In the Instance list, choose the item you want to analyze.

3 On the Association controls, in the Association area, click **Jobs**.

4 In the Association area, locate the jobs that have changed (indicated by a Change icon). Drill down to one of these jobs.

5 On the View controls in the Main area, click **Job Changes**.
Observe which changes were made to the job or to its steps and schedules. Determine if the job change has affected performance.

6 For example, if you notice that a job's schedule was changed do the following:

■ On the View controls in the Main area, click **Overview**.

■ Check the In MS-SQL graph to see how the selected job behaved in the SQL Server over the specified time period.

Examining the last execution of a job

Examining a job's last run can help you detect job failure and abnormal duration. Drilling down to the specified job run can help you learn more about its run by viewing the steps chain and the status and duration of each step.

To examine the last execution of a job

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

2 In the Instance list, choose the item you want to analyze.

3 On the Association controls, in the Association area, click **Jobs**.

4 On the Last Run tab, drill down to the job you want to analyze.

5 On the View controls in the Main area, click **Overview**.

6 View the steps chain and the status and duration of each step

Monitoring the occurrences of alerts

Monitoring alerts can help identify when an alert last occurred, what notifications were defined on the alert, when the notifications where last sent and which operators were involved.

To monitor the occurrences of alerts

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

2 In the Instance list, choose the item you want to analyze.

3 On the Association controls, in the Association area, click **Alerts**.

4 On the Details tab view when the alert last occurred.

5 Drill down to an alert to observe the actions performed when the alert was triggered and whether those actions were responsible for bottlenecks.

Monitoring a database (DB) maintenance plan

Monitoring a database (DB) maintenance plan can help you learn about the actions being performed in each maintenance plan and detect performance problems resulting from these actions. Drill down to a DB Maintenance Plan entity and choose the Performance view to analyze a plan's duration over time and detect problematic periods.

To monitor a database (DB) maintenance plan

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

2   In the Instance list, choose the item you want to analyze.

3   On the Association controls, in the Association area, click **DB Maintenance Plan.**

4   Drill down to a DB Maintenance Plan entity.

5   On the View controls in the Main area, click **Performance**. Analyze how long the plan ran, during the selected time period, and try to detect any problematic periods.

6   On the View controls in the Main area, click **History**. Observe the actions performed in the specified plan.

7   Check the jobs listed in the Association area to observe how each job contributed to the entire plan and drill down to a specific job to focus on a performance problem.

12

Pinpointing problems in a database

This section includes the following topics:

About the SmarTune tab

The SmarTune process detects performance deterioration by regularly analyzing SQL Server performance data, and identifying less than optimal behavior.

Note: You can change the frequency of how often the processes are run and configure the process parameters in AdminPoint. See the *Precise Administration Guide* for details.

SmarTune displays performance findings that may indicate potential problems in the database. The list of problems includes the following:

- list of statements that require tuning
- list of objects displaying heavy activity or possessing an inefficient structure
- instance events

This is the tab to begin with, if you want to start performance tuning on a database. In the SmarTune tab, you can examine the SmarTune findings, read expert knowledge regarding the area of deterioration, and obtain comprehensive advice on how to solve a specific problem and optimize performance. The Findings area lists the problematic statements, objects, and instance events that are causing poor performance. Relevant information regarding each problem listed in the Findings area is displayed in the right pane.

As part of the guidance provided, SmarTune includes Quick Links to launch to other tabs in context, so that you can easily continue your investigation. For example, you may be directed to the Activity tab to view the performance of an offending SQL statement. From there, you can launch to the SQL tab to examine the statement's execution plan and view a recommendation on a new index.

About accessing the SmarTune tab

You can access the SmarTune tab from the following locations:

- From the Dashboard

  Select the SmarTune icon in the Findings table, for a given instance. The SmarTune tab opens with the selected instance, displaying the findings for this instance, in the context of the selected instance and timeframe.

- From the Activity tab

Click SmarTune on the Tab Selection bar.

The SmarTune tab opens with the selected instance, displaying the findings for this instance, in the context of the selected instance and timeframe.

- From the Current, Objects, or Maintenance tabs

Click SmarTune on the Tab Selection bar.

The SmarTune tab opens with the selected instance, displaying the findings for this instance, in the context of the selected instance.

How the Precise for SQL Server SmarTune tab is structured

The SmarTune tab is divided into two areas—the Findings area and the Details area. The Findings area displays all findings returned by SmarTune for the selected time frame. The Details area provides detailed information regarding the selected finding.

About the Findings area in the SmarTune tab

The Findings area lists all the statements, collapsed statements, tables or instance events that were detected for a specific instance, within the selected time frame. All problems are ranked by severity. An Ignore icon, lets you decide if you want to remove a problem from the Findings table. This does not mean that SmarTune will stop checking these findings, rather that they will not be displayed in the Findings table. Clicking on an item in the Findings list refreshes the information displayed in the Details area.

The following table shows the information displayed in the Instance table.

Table 12-1 Findings table

Column	Description
Severity icon	Indicates how severe the detected problem really is. Calculates the severity of the finding in relation to the entire instance.
Type	Indicates the type of finding. Can be one of the following: ■ Heavy statement ■ Heavily accessed object ■ Heavy collapsed statement ■ Instance event
Object	Depending on the type of finding, displays one of the following: ■ For a heavy statement: the statement ID ■ For a heavily accessed object: the table name ■ For a heavy collapsed statement: the collapsed statement ID ■ For an instance event: empty field
In MS-SQL	Depending on the type of finding, displays one of the following: ■ For a statement: total time spent In MS-SQL, during the selected time period. ■ For an object: total In MS-SQL contribution to the table, during the selected time period. ■ For a collapsed statement: total In MS-SQL time of all statements that are part of the collapsed statement. ■ For an instance event: the In MS-SQL time related to the event, when available.
Database	Displays the name of the statement or object affected by the finding. The field is left empty in the case of an instance event finding.

Table 12-1   Findings table

Column	Description
User	Depending upon the type of finding, displays one of the following: ■ for a statement: parsing user ■ for an object: owner ■ for an instance event: empty field
Last Detected	Displays the last time the finding was found.
Ignore icon	Click to ignore a finding. The finding is added to an Ignore list and is no longer displayed in the Findings table. You can always redisplay a finding by turning off the Ignore list or by specifying which findings you want to restore in the Ignore list.

The following table shows the icons that signify the level of severity of a finding in the Findings table.

Table 12-2   About severity icons

Severity icon	Level of severity
	Low
	Medium
	High

About the Details area in the SmarTune tab

The Details area displays comprehensive information on the problem selected in the Findings area. A set of tabs lets you page through different views to gain additional information on the selected finding. The types of tabs, and the information displayed in them, changes according to the type of finding you chose to investigate in the Findings area.

The following table shows the information displayed for problematic statements.

Table 12-3   Information displayed in the Details area when a statement is selected

Tab	Information displayed
Highlights	Offers a quick overview of the major finding details. The Text area displays the actual text of the statement. The Statement Findings area presents an overview of all the findings detected within a time frame, for the selected statement. ■ Severity icon: indicates how serious a problem is, in relation to the statement's activity. ■ Launch icon: lets you launch, in context, to a different tab to continue your analysis. The tab you launch to depends upon SmarTune's recommendations. ■ Type: shows a description of the finding. ■ Table Name: displays the name of the table the finding is related to. ■ Potential Gain: calculates the possible gain obtained if the suggested change is implemented. ■ Last Detected: last time a problem was found for the specific finding. The Details for Finding area points you in the direction of additional tuning and findings-related data. They direct you to other tabs that can supply additional information regarding the specific finding, or to additional views within the SmarTune tab, so that you can view additional information about the specific finding.
Objects	Displays the list of objects appearing in a statement, that are part of the performance finding. For example, Objects area can indicate if a table or index is part of the heavy operators finding, or if a table has missing indexes or statistics.

Table 12-3 Information displayed in the Details area when a statement is selected

Tab	Information displayed

Over Time	Displays the following overtime graphs:
	
	■ Statement In MS-SQL
	    Displays the amount of time a statement spent in MS-SQL, during the selected time period.
	■ Statement Scalability
	    Displays the number of times a statement was executed, over the selected time period.
	■ Avg. Duration vs. Schema Changes
	    Displays the types of changes made to the schema, during the selected time period, such as whether an entity was dropped, created, or changed. This enables you to analyze how a schema change affected resource consumption and system performance.

The following table shows the information displayed for problematic objects.

Table 12-4 Information displayed in the Details area when an object is selected

Tab	Information displayed
Highlights	Offers a quick overview of the major finding details, such as the table's details, and a description of the findings, including an estimate of the potential improvement to performance if you choose to make the suggested change. ■ Severity icon: indicates how serious a problem is, in relation to the statement's activity. ■ Launch icon: lets you launch, in context, to a different tab to continue your analysis. The tab you launch to depends upon SmarTune's recommendations. ■ Type: shows a description of the finding. ■ Potential Gain: calculates the possible gain obtained if the suggested change is implemented. ■ Last Detected: last time a problem was found for the specific finding. The Details for Finding area points you in the direction of additional tuning and findings-related data. They direct you to other tabs that can supply additional information regarding the specific finding, or to additional views within the SmarTune tab, so that you can view additional information about the specific finding.
Statements	Displays a list of statements that accessed a table, during the selected time period.

Over Time	Displays the following overtime graphs:
	
	■ Storage
	    Displays an object's space usage, during the selected time period.
	■ Table In MS-SQL
	    Displays the MS-SQL contribution of the table, during the selected time period.

The following table shows the information displayed for problematic objects.

Table 12-5 Information displayed in the Details area when a collapsed statement is selected

Tab	Information displayed
Highlights	The Text area displays the actual text of the collapsed statement. The Collapsed Statement Findings area presents an overview of all the findings detected within a time frame, for the selected collapsed statement. ■ Severity icon: indicates how serious a problem is, in relation to the collapsed statement's activity. ■ Launch icon: lets you launch, in context, to a different tab to continue your analysis. The tab you launch to depends upon SmarTune's recommendations. ■ Type: shows a description of the finding. ■ Table Name: displays the name of the table the finding is related to. ■ Potential Gain: calculates the possible gain obtained if the suggested change is implemented. ■ Last Detected: last time a problem was found for the specific finding. The Details for Finding area points you in the direction of additional tuning and findings-related data. They direct you to other tabs that can supply additional information regarding the specific finding, or to additional views within the SmarTune tab, so that you can view additional information about the specific finding.
Statements	Displays the list of statements belonging to the same collapsed statement, that were executed during the selected time period.

Over Time	Displays the following overtime graphs:
	
	■ Statement In MS-SQL
	  Displays the amount of time a statement spent in MS-SQL, during the selected time period.
	■ Statement Scalability
	  Displays the number of times a statement was executed, over the selected time period.

The following table shows the information displayed on findings, at the instance level. Findings at the instance level check general behavior and statistics counters and give a general indication of the problems influencing instance performance.

Table 12-6 Information displayed in the Details area when an instance is selected

Tab	Information displayed
Highlights	Offers a quick overview of the major finding details. The text area describes the problem, displays thresholds and provides links to other tabs that can supply additional information regarding the specific finding, or to additional views within the SmarTune tab, so that you can view additional information about the specific finding. The graph presents information that is relevant to the specific finding or instance event.
Activities	For: Locked Instance ■ Table displaying information on top 10 blocked statements ■ Instance activity overtime graph For: Tempdb bottleneck ■ Table displaying information on top 10 problematic statements ■ Tempdb files I/O requests pie chart ■ Tempdb I/O requests vs. other files I/O requests ■ Tempdb storage

Table 12-6   Information displayed in the Details area when an instance is selected

Tab	Information displayed
Statistics	For: The transaction log bottleneck ■ Table displaying information on top 10 statements suffering from transaction log bottlenecks. Click on a database in the table to launch it, in context, in the Statistics tab. ■ Table displaying top 10 statements waiting for log. For: Buffer cache is too small ■ Page life expectancy graph. Indicates the average number of seconds a page stays in cache. If this value falls below the threshold set in AdminPoint, this can indicate that SQL Server may need more memory to boost performance. ■ Lazy writes/sec graph. Tracks how many times a second that the Lazy Writer process moves dirty pages from the buffer to disk to free up buffer space and make them available to user processes. Should be close to zero. ■ Page faults/sec graph. Indicates the overall rate at which the processor handles both hard and soft page faults. For Tempdb bottleneck, three graphs showing tempdb statistics: ■ Temp tables creation rate ■ Temp tables for destruction ■ Active temp tables

About Statement findings

SmarTune displays the top statements occurring in a time slice. A SmarTune time slice is updated on an hourly basis. Top statements are the statements that have the highest In MS-SQL time, computed without lock time and remote wait time that were running more than a specified number of minutes, during the time slice.

You can configure these parameters (how long they should be running to be considered top statements) in AdminPoint.

You decide how many top statements to display.

The following types of findings on statement behavior are displayed:

- Heavy Statement
- Heavy Collapsed Statement
- Major Statement in Batch
- Heavy Operators
- Missing Indexes
- Missing Statistics
- Table Schema Change May Increase Its Accessing Time
- Statement Is Not Scalable
- Table Growth May Increase Its Accessing Time
- Increase in Resource Consumption
- Statement Was Locked
- Statement Activity Consistently High

Heavy Statement

The statement is a major consumer of MS-SQL resources. By tuning the statement, you can free resources needed by other statements and processes.

Major Statement in Batch

The statement consumed more than 50% of MS-SQL batch resources. By tuning the statement, you can free resources needed by other statements and processes.

Heavy Operators

The statement or batch has an operator that accesses the object and shows a high cost percentage.

Missing Indexes

During SQL Server optimization, missing indexes were identified for one or more statements or batch. This means there can be one or more indexes, but they are not used because of a mismatch of column types. The SQL Server optimizer recommended creating indexes to improve the performance of the statement or batch. Detailed information regarding the index recommendation for this statement or batch can be found in the Recommend tab of the SQL tab.

Missing Statistics

The Missing Statistics warning was issued by the SQL Server optimizer when the access plan was created. This means that the SQL Server optimizer recommends creating and updating the statistics on objects accessed by a specific statement or batch. Detailed information on missing statistics for this statement or batch can be found in the Recommend tab in the SQL tab.

Table Schema Change May Increase Its Accessing Time

The average In MS-SQL time increased after an object change. Check the changes and how they affected the statement or batch execution time.

Statement Is Not Scalable

Statement resource consumption was increased by n% as a result of an increase in its executions.

Table Growth May Increase Its Accessing Time

The average In MS-SQL time of the statement increased following a major change in table size.

Increase in Resource Consumption

The SQL Server resources consumed by the statement increased by n%.

Statement Was Locked

Much of the statement or batch time was spent waiting for a lock.

Regular locks can be categorized as follows:

- During the blocker session, a locked statement or batch ran for a short period of time. Afterwards the session was idle or continued running other statements or batches. In this case, it is possible to identify the blocker session, but not necessarily the blocker statement or batch.

- During the blocker session, a locked statement or batch ran for a long period of time. Identifying the blocker statement or batch is easier in this case.

Statement Activity Consistently High

Total In MS-SQL time of the statement was consistently high and reached the thresholds of the top statements.

About Collapsed Statement findings

A collapsed statement includes several statements that use the same text, but not the same constants.

Heavy Collapsed Statement

A set of collapsed statements constitute a major consumer of MS-SQL resources.

Statement Is Not Scalable

Statement resource consumption was increased by n% as a result of an increase in its executions.

About Object findings

Object findings are checked for top objects.

The parameters determining what is considered to be a top object can be configured in AdminPoint.

The following types of findings on object behavior are displayed:

- [Heavily Accessed Object](#)
- [Heavy Operators](#)
- [Missing Indexes](#)
- [Missing Statistics](#)
- [Table Schema Change May Increase Its Accessing Time](#)
- [Object Is Not Scalable](#)
- [Table Growth May Increase Its Accessing Time](#)
- [Increase in Resource Consumption](#)
- [Locked Object](#)
- [Index Overhead](#)

Heavily Accessed Object

Object is a major consumer of MS-SQL resources. By tuning the object, you can free resources needed by other statements and processes.

Heavy Operators

One or more statements access this object using heavy operators. Statement has major access plan operators for this object that scan a great deal of data and show a high cost percentage.

Missing Indexes

During SQL Server optimization, missing indexes were identified for one or more statements. This means that the SQL Server optimizer recommended creating and updating the statistics on objects accessed by a specific statement or batch. Detailed information regarding the index recommendation for this statement can be found in the Recommend tab of the Objects tab.

Missing Statistics

The Missing Statistics warning was issued by the SQL Server optimizer when the access plan was created. This means that the SQL Server optimizer recommends creating and updating the statistics on the object. Detailed information on missing statistics for this object can be found in the Recommend tab in the Object tab.

Table Schema Change May Increase Its Accessing Time

The total In MS-SQL time of the object increased after changes were made to the schema.

Object Is Not Scalable

The total In MS-SQL contribution of the object increased when the number of statements executions changed.

Table Growth May Increase Its Accessing Time

The total In MS-SQL time of the object increased following a major change in the table size.

Increase in Resource Consumption

The total In MS-SQL time of the object has increased.

Locked Object

Much of the In MS-SQL time was spent waiting for a lock on the table.

Regular locks can be categorized as follows:

- During the blocker session, a locked statement ran for a short period of time. Afterwards the session was idle or continued running other statements. In this case, it is possible to identify the blocker session, but not necessarily the blocker statement.

- During the blocker session, a locked statement ran for a long period of time. Identifying the blocker statement is easier in this case.

Index Overhead

Most of the activity on the index is due to the fetching of index pages from the disk, reflect5ng changes made by INSERT, DELETE, and UPDATE statements.

About Instance findings

SmarTune checks to which extent a finding affects an instance.

The following types of findings on instance behavior are displayed:

- [Locked Instance](#)

- [Tempdb Bottleneck](#)

- [Buffer Cache Is Too Small](#)

- [Other Applications Influence SQL Server (Memory)](#)

- [Other Applications Influence SQL Server (CPU)](#)

- [The Transaction Log Bottleneck](#)

- [Extensive Internal Wait](#)

Locked Instance

x% of the In MS-SQL time was spent waiting for locks.

Regular locks can be categorized as follows:

- During the blocker session, a locked statement or batch ran for a short period of time. Afterwards the session was idle or continued running other statements or batches. In this case, it is possible to identify the blocker session, but not necessarily the blocker statement or batch.

- During the blocker session, a locked statement or batch ran for a long period of time. Identifying the blocker statement or batch is easier in this case.

Tempdb Bottleneck

The instance is experiencing a bottleneck of the tempdb database. x% of the In MS SQL time was spent waiting for tempdb. The tempdb database is used for temporary storage for sorting, joining, and, in SQL Server 2005, for row versioning.

Buffer Cache Is Too Small

A SQL Server instance is configured to work with a small buffer cache. Your applications are experiencing x% of the In MS-SQL time for I/O wait. The application's overall performance is therefore not optimal.

Other Applications Influence SQL Server (Memory)

A SQL Server instance experienced many page faults and lack of memory, while other processes on the same server did not experience any page faults problem.

Other Applications Influence SQL Server (CPU)

A SQL Server instance experienced CPU shortage, as a result of other processes running on the server.

The Transaction Log Bottleneck

Some of the databases suffer from transaction log bottleneck. Log wait consumes x% of the In MS-SQL time.

Extensive Internal Wait

The SQL Server instance has spent much of its time waiting for Internal Wait.

Configuring Findings settings

You determine the maximum number of findings to display in the Findings area.

Findings are added to the Findings table as they are detected by the SmarTune process. You can choose to ignore a finding by selecting the finding and clicking its ignore icon. Findings which you have decided to ignore can be returned to the Findings display by configuring the SmarTune settings.

Note: The SmarTune settings control the findings displayed in both the Dashboard and SmarTune tabs.

To determine the maximum number of findings to display

1   On the Settings menu, click **SmarTune Settings.**

2   In the SmarTune Settings dialog box, in the Maximum number of findings to display, select the number of findings you want to display.

3   Click **OK.**

To restore the display of a finding to the Findings table in the SmarTune tab

1   On the Settings menu, click **SmarTune Settings**.

2   In the SmarTune Settings dialog box, check the **Ignore** check box located next to the finding you want to ignore.

3   Check the **Enable Ignore List** check box.

4   Click **OK**.

To restore all findings detected by the SmarTune process to the Findings table in the SmarTune tab

1   On the Settings menu, click **SmarTune Settings**.

2   In the SmarTune Settings dialog box, clear the **Activate Ignore List** check box.

3   Click **OK**.

13

SQL Server Findings

This section includes the following topics:

About the Findings view

The Findings view displays recommendations that can be used to create a better execution plan and improve the performance of a statement. The Findings vary according to the type of operation in a statement, the Precise product and the technology.

About the Highlights area

The Highlights area displays a brief description of the findings for the selected type of operation.

About the What to do next area

The What to do next area displays one or more recommended steps to identify the cause of the problem. Carefully review all data for the finding before continuing.

About the Advice area

The Advice area displays one or more recommended options to resolve or reduce the problem for the selected finding. Carefully review all data for the finding and then perform the advice that best suits your needs.

How to investigate findings

When you start investigating the findings, it is good practice to start with the finding that has the highest severity ranking in the Findings table.

To investigate a finding

1 Identify the finding with the highest severity ranking in the Findings table.

2   Select the finding type to view additional information on the selected type of operation.

3   Read the Highlights and What to do next areas for the finding.

4   After you have studied all of the information provided, read the Advice area and perform the recommendation that best suits your needs.

5   Follow up on performance to verify that the problem was resolved.

About Statement findings

The following statement findings can help you tune your system:

- Heavy Statement
- Heavy Collapsed Statement
- Major Statement in Batch
- Heavy Operators
- Missing Indexes
- Missing Statistics
- Table Schema Change May Increase Its Accessing Time
- Statement Is Not Scalable
- Table Growth May Increase Its Accessing Time
- Increase in Resource Consumption
- Statement or Batch Was Locked
- Statement Activity Consistently High

Heavy Statement

The statement is a major consumer of MS-SQL resources. By tuning the statement, you can free resources needed by other statements and processes.

Table 13-1   Heavy Statement findings

Column	Description
What to do next	Try to determine what is causing the statement's high resource consumption. In the SQL tab, examine the text of the relevant statement, and its findings, execution plan, change data and statistics.
Advice	The following scenarios indicate what factors can lead to heavy resource consumption and what steps you can take: ■ A heavy operator was identified. Use the SQL findings to identify the heaviest operator. ■ A missing index was identified (SQL Server 2005 only). The SQL Server optimizer identified missing indexes when the access plan was created. Detailed index information can be found in the Recommend tab in the SQL tab. ■ Missing statistics were identified. The SQL Server optimizer identified missing statistics when the access plan was created. Detailed information on missing statistics for this statement can be found in the Recommend tab in the SQL tab. ■ An increase of In MS-SQL time was identified. Use Precise for SQL Server to try and locate the cause. Check schema changes, major table growth and scalability changes.

Heavy Collapsed Statement

A collapsed statement includes several statements that use the same text, but not the same constants.

A set of collapsed statements constitute a major consumer of MS-SQL resources. By tuning the set of statements, you can free resources needed by other statements and processes.

Table 13-2 Heavy collapsed statement findings

Column	Description
What to do next	In the SmarTune tab, examine the text of the relevant statements. Check their scalability and associated list of statements. Select the statement you want to analyze, and launch to the SQL tab with the statement in context. Examine its execution plan, change data and statistics to determine what is causing its high resource consumption.
Advice	The following scenarios indicate what factors can lead to heavy resource consumption and what steps you can take: ■ A scalability issue was raised. Check performance over time in the Over Time tab. ■ In the Statements tab, tune a statement by doing one of the following: ■ If a heavy operator was identified, use the SQL findings to identify the heaviest operator. ■ If a missing index was identified (SQL Server 2005 only), this means that the SQL Server optimizer identified a missing index when the access plan was created. Detailed index information can be found in the Recommend tab in the SQL tab. ■ If missing statistics were identified this means that the SQL Server optimizer identified missing statistics when the access plan was created. Detailed information on missing statistics for the collapsed statement can be found in the Recommend tab in the SQL tab. ■ If an increase in MS-SQL time was identified, use Precise for SQL Server to try and locate the cause. Check schema changes, major table growth and scalability changes.

Major Statement in Batch

The statement consumed more than 50% of MS-SQL batch resources. By tuning the statement, you can free resources needed by other statements and processes.

Table 13-3 Major Statement in Batch findings

Column	Description
What to do next	Tune the specific statement and examine its text, findings, execution plan, change data and statistics to determine what is causing the statement's high resource consumption.
Advice	The following scenarios indicate what factors can lead to heavy resource consumption and what steps you can take: A heavy operator was identified. Use the SQL findings to identify the heaviest operator. A missing index was identified (SQL Server 2005 only). The SQL Server optimizer identified missing indexes when the access plan was created. Detailed index information can be found in the Recommend tab in the SQL tab. Missing statistics were identified. The SQL Server optimizer identified missing statistics when the access plan was created. Detailed information on missing statistics for this statement can be found in the Recommend tab in the SQL tab. An increase of In MS-SQL time was identified. Use Precise for SQL Server to try and locate the cause. Check schema changes, major table growth and scalability changes.

Heavy Operators

The statement or batch has an operator that shows a high cost percentage.

Table 13-4 Heavy Operators findings

Column	Description
What to do next	Perform one of the following options: ■ Click **Locate** to find the heavy operator and tune it. ■ Identify recommendations in the Recommend tab in the SQL tab. ■ In case of **Hash** or **Sort** operators examine internal wait for the statement.

Table 13-4 Heavy Operators findings

Column	Description
Advice	One of the following operations (for example, table scan, index scan, clustered index scan or filter) was identified as a major cost indicator. Try to eliminate the major consuming operation by doing one of the following:
	For a table scan:
	■ Eliminate a full table scan or cluster index scan.
	■ Try to identify high selectivity columns for the statements or batches. Create an index matching the statement or batch's predicates.
	■ Try to identify if there is a small set of columns that can be used in an index-only operation.
	■ Partition the table according to the best predicates existing in the statements or batches.
	■ Improve full table scan. Table fragmentation can affect overhead when a table scan or partial table scan is performed. Determine whether the table is heavily fragmented. If it is, defragment the table.
	For an index scan:
	■ If a missing index finding is detected, launch to the Recommend tab. Otherwise try to identify where the matching level problem exists, indicating that the index column order does not match the statement or batch WHERE columns.
	Example:
	■ An index on A, B, C, while the statement or batch is "select … where A=3 and C=7".
	■ Many irrelevant index leaf pages will be read in this case. This can be avoided by either changing the index column sequence or adding columns to the index.
	■ Analyze the recommendations provided in the Recommend tab in the SQL tab.
	For a Filter:
	Eliminate a filter by doing one of the following:
	■ Consider creating an index view.
	■ Consider performing an index on the computed column.
	For a Sort:
	■ Try reducing the space or memory required for Sort operations by reducing the number of sorted columns or by filtering the rows to be sorted.
	■ Try to find a way to eliminate the sort by using pre-sorted information like creating an index on the sorted columns.
	■ Check the values of the *Minimum memory per query (KB)* and *Minimum memory for index create sorts (KB)* instance parameters. Use the *memory for index create sorts (KB)* configuration to control the amount of memory used by index creation sorts. The *Minimum memory for index create sorts (KB)* configuration is self-configuring and should work in most cases without requiring adjustment. However, if you experience difficulties creating indexes, consider increasing the value of this option from its run value. Query sorts are controlled through the *Minimum memory per query (KB)*.
	For a Hash:
	Hash join is used where the joined tables or row sets are large or where there aren't adequate indexes or the lack of indexes at all.
	■ Verify that there are **no missing** indexes or the indexes are **adequate**.
	■ Search for a missing **WHERE** clause or a missing condition in the WHERE clause.
	■ Search for a **non-sargeable** expression. A **non-sargeable** expression is an expression preventing the optimizer of using the index in the ideal way or not at all like a function expression.
	■ Check the values of the *Minimum Size of Server Memory (MB)* and *Max Size of Server Memory (MB)* instance parameters.
	For a Index Spool, Table Spool and Row Count Spool:
	Spooling are internally temporary tables created by SQL Server on the tempdb database. Spooling may cause tempdb overhead and thus resulting in instance performance degradation caused by tempdb wait or tempdb major growth.
	■ Try to eliminate spooling operations by rephrasing the query.
	■ Try to reduce the number of logical reads or writes.

Missing Indexes

During SQL Server optimization, missing indexes were identified for the statement or batch. This means there can be one or more indexes, but they are not used because of a mismatch of column types. The SQL Server optimizer recommended creating indexes to improve the performance of the statement or batch. Detailed information regarding index recommendation for this statement or batch can be found in the Recommend tab of the SQL tab.

Table 13-5   Missing Indexes findings

Column	Description
What to do next	Examine the information displayed in the Recommend tab in the SQL tab to identify missing statistics and analyze index recommendations.
Advice	Perform one of the following options: ■ Examine the information displayed in the Recommend tab in the SQL tab to identify missing statistics and analyze index recommendations. ■ Use the What if feature in the Recommend tab to evaluate which statements or batches were affected, based on a large set of statements or batches that were active during the selected time frame. This feature lets you check whether it is possible to improve performance by creating one or several indexes.

Missing Statistics

The Missing Statistics warning was issued by the SQL Server optimizer when the access plan was created. This means that the SQL Server optimizer recommends creating and updating the statistics on objects accessed by a specific statement or batch. Detailed information on missing statistics for this statement or batch can be found in the Recommend tab in the SQL tab.

Table 13-6   Missing Statistics findings

Column	Description
What to do next	Examine the information displayed in the Recommend tab in the SQL tab to identify missing statistics and analyze index recommendations.
Advice	Perform the following options: ■ Examine the information displayed in the Recommend tab in the SQL tab to identify missing statistics and analyze index recommendations. ■ Use the What if feature in the Recommend tab to evaluate a missing index, based upon a large set of statements or batches that were active during the selected time frame. ■ This feature lets you check whether it is possible to improve performance by creating one or several indexes. ■ Once statistics are collected, remember that statistics should be periodically maintained.

Table Schema Change May Increase Its Accessing Time

The average In MS-SQL time increased after an object change. Check the changes and how they affected the statement execution time.

Table 13-7   Table Schema Change May Increase Its Accessing Time findings

Column	Description
What to do next	In the SQL tab, examine the information displayed in the History tab to identify object changes and correlate these changes to performance degradation.
Advice	Several changes may effect In MS-SQL time and may effect other statements In MS-SQL as well. Changes such as index creation/drop can cause access plan changes that can effect statement performance. Do the following: ■ Identify the change in the History tab ■ Compare statement or batch execution plans in the Compare tab.

Statement Is Not Scalable

Statement resource consumption was increased by n% as a result of an increase in its executions.

Table 13-8   Statement Is Not Scalable findings

Column	Description
What to do next	Perform one of the following options: ■ Examine scalability by analyzing degradation in average duration for statement resulting from an increase in the number of executions. ■ Check the program executing the statement and try to identify application changes. ■ Explore other statement or batch execution identifiers, such as, machine or users, in the Activity tab.
Advice	Perform the following options: ■ Try to determine if this is a pure scalability issue. Increase the time frame and explore statement scalability over a larger period of time. If it is a problem of scalability, this can be treated in many ways. ■ For example, you can avoid unscalable operations, such as table scans, in an execution plan. ■ Eliminate concurrency by distributing the operation of the application over several MS-SQL servers.

Table Growth May Increase Its Accessing Time

The average In MS-SQL time of the statement increased following a major change in table size.

Table 13-9   Table Growth May Increase Its Accessing Time findings

Column	Description
What to do next	Examine the information displayed in the History tab in the SQL tab to identify volume changes and correlate these changes to performance degradation.
Advice	Volume changes may effect response time especially in: ■ Table scans: may effect costs dramatically and create larger scans. ■ Index scans: may create larger scans. ■ Index structure: Every level in the index depth requires I/O synchronization. An index depth of four will result in four synchronized I/Os for each key, meaning that I/O cost will be too high and the optimizer may choose not to use the index. Try defragmenting the index or redesigning it. ■ Examine changes in the History tab and analyze these changes over time by launching to the Objects tab.

Increase in Resource Consumption

The SQL Server resources consumed by the statement increased by n%.

Table 13-10   Increase in Resource Consumption findings

Column	Description
What to do next	Perform one of the following options: ■ Examine the history of the statement. ■ Examine the index and statistics recommendations for the selected statement. ■ Examine the statement plan in the SQL tab. ■ Examine statement activity in the Activity tab.
Advice	The average In MS-SQL time of the statement or batch has increased but not as a result of major table growth, scalability change or schema changes. A major change can result from a change in the execution plan. Examine the history of the statement and try to identify whether a major change took place. Compare explain plans using the Compare tab. Check instance behavior. An increase in instance execution over time, may affect the performance of regular statements.

Statement or Batch Was Locked

Much of the statement or batch time was spent waiting for a lock.

Regular locks can be categorized as follows:

- During the blocker session, a locked statement or batch ran for a short period of time. Afterwards the session was idle or continued running other statements or batches. In this case, it is possible to identify the blocker session, but not necessarily the blocker statement or batch.

- During the blocker session, a locked statement or batch ran for a long period of time. Identifying the blocker statement or batch is easier in this case.

Table 13-11 Statement or Batch Was Locked findings

Column	Description
What to do next	Perform one of the following options: ■ Check the statement or batch locking information and associated blocker sessions in the Activity tab. ■ Check the locked object in the Activity tab. ■ Check statement or batch activity in the Activity tab. Increase the time frame, if necessary.
Advice	Identify the blocker session in the Activity tab and examine the application and application timing. Examine the lock chain to identify the statement or batch holding the lock: ■ In the Activity tab sort statements or batches by their lock wait. ■ Identify major statements or batches. ■ Narrow the time frame. Drill down into major statements or batches. ■ In the Association area, select Blocker Sessions. ■ Try to identify lock patterns within a larger time frame.

Statement Activity Consistently High

Total In MS-SQL time of the statement was consistently high and reached the thresholds of the top statements.

Table 13-12 Statement Activity Consistently High findings

Column	Description
What to do next	Perform one of the following solutions: ■ Examine statement activity in the Activity tab. ■ Examine the index and statistics recommendations for the selected statement. ■ Examine the plan of the statement in the SQL tab.
Advice	The In MS-SQL time of the statement is consistently high for the select time frame, as compared with resource consumption of the previous week. This finding is issued when the statement shows up within the top resource consumers of your application, their resource consumption was always high, and no major change in resource consumption occurred during the last two weeks. Check which resource is most-consumed by the statement or batch and determine how you can make it available to the statement. For example: Consider scheduling different activities that uses the same resource at different times, thereby freeing the resources for the statement.

About Object findings

The following object findings can help you tune your system:

- [Heavily Accessed Object](#)
- [Heavy Operators](#)
- [Missing Indexes](#)
- [Missing Statistics](#)
- [Table Schema Change May Increase Its Accessing Time](#)
- [Object Is Not Scalable](#)
- [Table Growth May Increase Its Accessing Time](#)
- [Increase in Resource Consumption](#)
- [Locked Object](#)
- [Index Overhead](#)
- [High Amount of Index Scans](#)
- [High Amount of Table Lookups](#)
- [High Amount of Unused Heap Pages](#)

Heavily Accessed Object

Object is a major consumer of MS-SQL resources. By tuning the object, you can free resources needed by other statements and processes.

Table 13-13   Heavily accessed object findings

Column	Description
What to do next	Perform one of the following options: ■ Examine the SmarTune object findings for the selected object. ■ Examine the index and statistics recommendations in the Recommend tab in the Objects tab. ■ Examine table details in the Objects tab. ■ Examine table activity in the Activity tab.
Advice	The following scenarios indicate which factors can lead to heavy resource consumption and what steps you can take: ■ In MS-SQL time of the object increased. Identify and try to find the source of the change. An example of such a change can be table growth or statistics changes. ■ A missing index was identified (SQL Server 2005 only). The SQL Server optimizer identified missing indexes when the access plan was created. Detailed index information can be found in the Recommend tab in the Objects tab. ■ Locked Object. Try to identify blocker sessions. heavy operators were identified. Check the information displayed in the Overview in the Statements tab in the SmarTune tab. ■ The object is suffering from high number of index scans. Try to identify where a matching level problem exists, indicating that the index column order does not match the statement or batch WHERE columns. Example: An index on A, B, C, while the statement or batch is "select … where A=3 and C=7". Many irrelevant index leaf pages will be read in this case. This can be avoided by either changing the index column sequence or adding columns to the index. Analyze the recommendations provided in the Recommend tab in the SQL tab. ■ The object is suffering from high number of table lookups. Verify that all the columns in the select list are indeed needed. Try to use the INCLUDE (2005 only) option to add the needed columns to the index leafs to prevent unnecessary table lookups or in case of 2000 instance, use the index covering technique to eliminate extra table lookups

Heavy Operators

One or more statements access this object using heavy operators. Statement has major access plan operators for this object that scan a great deal of data and show a high cost percentage.

Table 13-14   Heavy Operators findings

Column	Description
What to do next	Perform one of the following options: ■ Examine the index and statistics recommendations for this object in the Recommend tab, in the Objects tab. ■ Explore table details in the Objects tab. ■ Examine table activity in the Activity tab.

Table 13-14 Heavy Operators findings

Column	Description
Advice	One of the following operations (for example, table scan, index scan, clustered index scan or filter) was identified as a major cost indicator. Try to eliminate the major consuming operation by doing one of the following: For a table scan: ■ Eliminate a full table scan or cluster index scan. ■ Try to identify high selectivity columns for the statements or batches. Create an index matching the statement or batch's predicates. ■ Try to identify if there is a small set of columns that can be used in an index-only operation. ■ Partition the table according to the best predicates existing in the statements or batches. ■ Improve full table scan. Table fragmentation can affect overhead when a table scan or partial table scan is performed. Determine whether the table is heavily fragmented. If it is, defragment the table. For an index scan: ■ If a missing index finding is detected, launch to the Recommend tab. Otherwise try to identify where the matching level problem exists, indicating that the index column order does not match the statement or batch WHERE columns. Example: ■ An index on A, B, C, while the statement or batch is "select … where A=3 and C=7". ■ Many irrelevant index leaf pages will be read in this case. This can be avoided by either changing the index column sequence or adding columns to the index. ■ Analyze the recommendations provided in the Recommend tab in the SQL tab. For a Filter: Eliminate a filter by doing one of the following: ■ Consider creating an index view. ■ Consider performing an index on the computed column. For a Sort: ■ Try reducing the space or memory required for Sort operations by reducing the number of sorted columns or by filtering the rows to be sorted. ■ Try to find a way to eliminate the sort by using pre-sorted information like creating an index on the sorted columns. ■ Check the values of the *Minimum memory per query (KB)* and *Minimum memory for index create sorts (KB)* instance parameters. Use the *memory for index create sorts (KB)* configuration to control the amount of memory used by index creation sorts. The *Minimum memory for index create sorts (KB)* configuration is self-configuring and should work in most cases without requiring adjustment. However, if you experience difficulties creating indexes, consider increasing the value of this option from its run value. Query sorts are controlled through the *Minimum memory per query (KB)*. For a Hash: Hash join is used where the joined tables or row sets are large or where there aren't adequate indexes or the lack of indexes at all. ■ Verify that there are **no missing** indexes or the indexes are **adequate**. ■ Search for a missing **WHERE** clause or a missing condition in the WHERE clause. ■ Search for a **non-sargeable** expression. A **non-sargeable** expression is an expression preventing the optimizer of using the index in the ideal way or not at all like a function expression. ■ Check the values of the *Minimum Size of Server Memory (MB)* and *Max Size of Server Memory (MB)* instance parameters. For a Index Spool, Table Spool and Row Count Spool: Spooling are internally temporary tables created by SQL Server on the tempdb database. Spooling may cause tempdb overhead and thus resulting in instance performance degradation caused by tempdb wait or tempdb major growth. ■ Try to eliminate spooling operations by rephrasing the query. ■ Try to reduce the number of logical reads or writes.

Missing Indexes

During SQL Server optimization, missing indexes were identified for one or more statements. This means that the SQL Server optimizer recommended creating indexes to improve the performance of the statement. Detailed information regarding index recommendation for this statement can be found in the Recommend tab of the Objects tab.

Table 13-15   Missing Indexes findings

Column	Description
What to do next	Examine the information displayed in the Recommend tab in the Objects tab to identify index recommendations.
Advice	Perform one of the following options: ■ Examine the information displayed in the Recommend tab in the Objects tab to identify index recommendations. ■ Use the What if feature in the Recommend tab to evaluate which statements were affected, based on a large set of statements that were active during the selected time frame. This feature lets you check whether it is possible to improve performance by creating one or several indexes.

Missing Statistics

The Missing Statistics warning was issued by the SQL Server optimizer when the access plan was created. This means that the SQL Server optimizer recommends creating and updating the statistics on the object. Detailed information on missing statistics for this object can be found in the Recommend tab in the Object tab.

Table 13-16   Missing Statistics findings

Column	Description
What to do next	Examine the information displayed in the Recommend tab in the Objects tab to identify missing statistics and analyze index recommendations.
Advice	Perform the following options: Examine the information displayed in the Recommend tab in the Objects tab to identify missing statistics and analyze index recommendations. Use the What if feature in the Recommend tab to evaluate a missing index, based upon a large set of statements or batches that were active during the selected time frame. This feature lets you check whether it is possible to improve performance by creating one or several indexes. Once statistics are collected, remember that statistics should be periodically maintained.

Table Schema Change May Increase Its Accessing Time

The total In MS-SQL time of the object increased after changes were made to the schema.

Table 13-17   Table Schema Change May Increase Its Accessing Time findings

Column	Description
What to do next	Examine table details and schema changes in the Objects tab.
Advice	Several changes may effect in MS-SQL response time and may effect other statement changes as well. Changes such as index creation/drop can affect statement performance: ■ Identify the change in the Objects tab. ■ Examine the change, when it was made, the performance changes related to the change, and identify any other changes in the object that may result from the change.

Object Is Not Scalable

The total In MS-SQL contribution of the object increased when the number of statements executions changed.

Table 13-18   Object Is Not Scalable findings

Column	Description
What to do next	Examine the object's behavior over time and explore the usage patterns of the statements.
Advice	Perform one of the following options: ■ Examine the object's behavior over time and explore the usage patterns of the statements. ■ Check whether usage patterns dramatically changed. If yes, consider improving them by indexing or partitioning. ■ Identify major users by associating to users or machines in the Association. Determine if this is acceptable for your system based on your knowledge of your system.

Table Growth May Increase Its Accessing Time

The total In MS-SQL time of the object increased following a major change in the table size.

Table 13-19   Table Growth May Increase Its Accessing Time findings

Column	Description
What to do next	Examine table details and associated statements in the Objects tab to identify statements that may be affected by table growth.
Advice	Volume changes may affect response time especially in: ■ Table scans: may affect costs dramatically. ■ Index structure: Every level in the index depth requires I/O synchronization. An index depth of four will result in four synchronized I/Os for each key, meaning that I/O cost will be too high and the optimizer may choose not to use the index. Try defragmenting the index or redesigning it. Try to identify the heaviest statements and tune them.

Increase in Resource Consumption

The total In MS-SQL time of the object has increased.

Table 13-20   Increase in Resource Consumption findings

Column	Description
What to do next	Perform one of the following options: Examine table details of common access patterns and associated statements in the Objects tab (SQL Server 2005 only). Examine the recommendations for the index and statistics of the selected object. Explore table details in the Objects tab.
Advice	The average In MS-SQL time of the object has increased but not as a result of major table growth, scalability change or schema changes. A major change can result from a change in execution plan. Examine the table details and the statement accessing it and try to identify what caused the change. Compare explain plans using the Compare tab.

Locked Object

Much of the In MS-SQL time was spent waiting for a lock on the table.

Regular locks can be categorized as follows:

- During the blocker session, a locked statement ran for a short period of time. Afterwards the session was idle or continued running other statements. In this case, it is possible to identify the blocker session, but not necessarily the blocker statement.

- During the blocker session, a locked statement ran for a long period of time. Identifying the blocker statement is easier in this case.

Table 13-21 Locked Object findings

Column	Description
What to do next	Perform one of the following options: ■ Examine locking information in the Activity tab. ■ Examine the index and statistics recommendations for the selected object. ■ Explore table details in the Objects tab.
Advice	Identify the blocker session in the Activity tab and examine the application and application timing. Examine the lock chain to identify the statement holding the lock.

Index Overhead

Most of the activity on the index is due to the fetching of index pages from the disk, reflecting changes made by INSERT, DELETE, and UPDATE statements.

Table 13-22 Index Overhead findings

Column	Description
What to do next	Perform one of the following options: ■ Examine the statements causing index updates, in the Statements tab. ■ Check if the index is being used in execution plans. ■ Try to identify index update patterns (such as, daily or nightly), in the In Oracle graph, in the Read/Write Operations tab.
Advice	Perform one of the following options: ■ When insert statements are part of a load, batch, or night activity, consider dropping the index before performing the activity, and recreating it afterwards. ■ If the index is not used in execution plans consider dropping the index or unused columns from the index, to reduce index overhead. If the index is used in execution plans, launch to the What-If tab to see which statements may be effected by this change.

High Amount of Index Scans

The index is mainly used for scans and not for seeks which is the recommended method.

Table 13-23 High Amount of Index Scan findings

Column	Description
What to do next	■ Examine the information displayed in the 'Common Access Patterns' and Operational Statistics' ■ Associate to statements ■ Examine the information displayed in the Recommend tab in the SQL tab to identify missing statistics and analyze index recommendations.

Table 13-23   High Amount of Index Scan findings

Column	Description
Advice	Perform one of the following options: ■ Examine the index scans percentage of overall usage of the index ■ Associate to statements and find out why the optimizer chose to use index scans by looking for missing predicates, functions on the index columns, non-selective columns, range scan queries, inadequate indexes, etc. ■ Examine the information displayed in the Recommend tab in the SQL tab to identify missing statistics and analyze index recommendations. ■ Use the What If feature in the Recommend tab to evaluate which statements or batches were affected, based on a large set of statements or batches that were active during the selected time frame. This feature lets you check whether it is possible to improve performance by creating one or several indexes.

High Amount of Table Lookups

The table has a high amount of lookups which can cause excessive I/O wait.

Table 13-24   High Amount of Table Lookup findings

Column	Description
What to do next	■ Examine the information displayed in the Recommend tab in the SQL tab to identify and analyze index covering recommendations.
Advice	Perform one of the following options: ■ Examine the information displayed in the Recommend tab in the SQL tab and check for index covering recommendations. ■ Use the What If feature in the Recommend tab to evaluate which statements or batches were affected, based on a large set of statements or batches that were active during the selected time frame. This feature lets you check whether it is possible to improve performance by creating one or several indexes.

High Amount of Unused Heap Pages

SQL Server reads a high amount of unused Heap Pages, which can lead to unnecessary I/O Wait.

Table 13-25   High amount of unused Heap Pages

Column	Description
What to do next	■ Examine the difference between the number of rows and allocated pages and consider rearranging the table by creating a cluster index and then dropping it.

About Instance findings

The following instance findings can help you tune your system:

- Locked Instance
- Tempdb Bottleneck
- Buffer Cache Is Too Small
- Other Applications Influence SQL Server (Memory)
- Other Applications Influence SQL Server (CPU)
- Transaction Log Bottleneck
- Extensive Internal Wait
- Tempdb Major Growth
- High CPU Wait

Locked Instance

x% of the In MS-SQL time was spent waiting for locks.

Regular locks can be categorized as follows:

- During the blocker session, a locked statement or batch ran for a short period of time. Afterwards the session was idle or continued running other statements or batches. In this case, it is possible to identify the blocker session, but not necessarily the blocker statement or batch.
- During the blocker session, a locked statement or batch ran for a long period of time. Identifying the blocker statement or batch is easier in this case.

Table 13-26   Locked Instance findings

Column	Description
What to do next	Perform one of the following options: ■ Examine the table activity and locking information in the Activity tab ■ Examine lock counters in the Statistics tab.
Advice	To reduce the lock wait for the instance, consider the following solutions: Concentrate on locked statements: ■ In the Activity tab sort statements by their lock wait. ■ Identify a major statement. ■ Narrow down the time frame and drill down into it. ■ Change association to Blocker Sessions. ■ Concentrate on locked objects. Check to see if the lock appears in the Current tab. If yes, examine the lock chain to identify the statement holding the lock. ■ Concentrate on blocker sessions. Try to identify the locking statement in the Activity tab using a narrower time frame matching, the lock periods. Focus on the locked table and associated statements. The DML statement (and update queries) that are NOT waiting for locks can be the immediate suspects.

Tempdb Bottleneck

The instance is experiencing a bottleneck of the tempdb database. x% of the In MS SQL time was spent waiting for tempdb. The tempdb database is used for temporary storage for sorting, joining, and, in SQL Server 2005, for row versioning.

Table 13-27 Tempdb bottleneck findings

Column	Description
What to do next	■ Examine the type of statements heavily using the tempdb resource in the Activity tab. ■ Check that a sort is being performed on the statements and tune the Sort operation. Tuning a Sort operation can be performed by making sure that only the relevant columns and rows are being sorted. You can view the columns being sorted in the SQL tab. ■ Check how the tempdb files are created in the Objects tab. For example, check on which devices the files reside and check the auto-extend parameters (a small value can create fragmentation at the OS level). ■ Check the load on the devices on which the tempdb files are created. You can view this in the Statistics tab.
Advice	■ Tune sorts or joins: Try reducing the space required for Sort operations by reducing the number of sorted columns or by filtering the rows to be sorted. Verify that Hash and Merge joins occurred as the result of the proper join methodology. Hash and Merge joins are temporary storage consumers. Create indexes to eliminate the sorts or change the join methodology to a Nested Loop. Check the values of the `min server memory (MB)` and `index create memory (KB)` instance parameters. ■ Tune tempdb files: Verify that tempdb files are distributed across several devices. Check the load on each device. Verify the auto-extend parameters. Verify that the file is not fragmented at the OS level. ■ Tune row versioning: Verify that row versioning is enabled only when needed. Verify that the Snapshot Isolation level is used properly. ■ Examine tempdb usage when the applications explicitly create temporary tables. Try creating indexes on those tables after they are populated.

Buffer Cache Is Too Small

A SQL Server instance is configured to work with a small buffer cache. Your applications are experiencing x% of the In MS-SQL time for I/O wait. The application's overall performance is therefore not optimal.

Table 13-28 Buffer Cache Is Too Small findings

Column	Description
What to do next	■ Examine the hit ratio of the caches in SQL Server, in the Statistics tab. ■ Examine the page faults the SQL Server issues. A small number of page faults and a low hit ratio can indicate that the SQL Server is not maximizing the use of server memory. ■ Examine the memory being allocated to the SQL Server as opposed to the memory available to the server. ■ Examine the memory consumption of other processes in the Insight Savvy for OS. ■ Examine how long the applications wait for I/O. ■ Examine page life expectancy, in the Statistics tab.
Advice	Before increasing memory for the SQL Server, check that you are not going to adversely affect other applications running on the server by doing so. In addition, adding increasing the SQL Server memory too much, can decrease the performance, since SQL Server will have to do paging. Experts recommend allowing operating systems to have 20% available memory. If you have other applications running on the same server, check their memory requirements in the Insight Savvy for OS. When you determine how much memory should be allocated to SQL Server, you can change its settings. In this case you can configure memory to be fixed and not dynamic. Check the value of the min server memory (MB) instance parameter. This parameter indicates how much memory is allocated to every session connected to the SQL Server. If you find that you are experiencing many idle sessions (due to the threshold settings in the connection pooling) you are wasting too much memory. SQL Server does not allow the configuration of different settings for each cache, so you must verify whether or not the buffer cache has enough memory.

Other Applications Influence SQL Server (Memory)

A SQL Server instance experienced many page faults and lack of memory, while other processes on the same server did not experience any page faults problem.

Table 13-29   Other applications influence SQL Server (memory) findings

Column	Description
What to do next	■ Examine the SQL Server page faults, in the Statistics tab. ■ Examine the memory allocated to SQL Server. ■ Examine the memory resources used by other processes on the server. ■ If Insight Savvy for OS is installed, check the memory usage of all processes on the server.
Advice	Since SQL Server is your RDBMS, and it is the single resource being used by most of your applications, it is essential that it receive all the resources it requires. You should be aware of which applications are also running on the server and how they affect your SQL Server. Examine the amount of memory being used by the SQL Server and the amount of memory being used by the other processes located on the server. Verify that SQL Server has enough memory allocated to it. Prioritize the other applications running on the server and decide if any of the other processes can be migrated to another server, thereby freeing memory resources to the SQL Server.

Other Applications Influence SQL Server (CPU)

A SQL Server instance experienced CPU shortage, as a result of other processes running on the server.

Table 13-30   Other applications influence SQL Server (CPU) resources findings

Column	Description
What to do next	■ Examine CPU usage of the server, in the Statistics tab. ■ Examine In MS-SQL breakdown, in the Activity tab. ■ If Insight Savvy for OS is installed, examine CPU usage of all processes on the server.
Advice	Since SQL Server is your RDBMS, and it is the single resource being used by most of your applications, it is essential that it receive all the resources it requires. You should be aware of which applications are also running on the server and how they affect your SQL Server. Examine the amount of CPU being used by the SQL Server and the amount of memory being used by the other processes located on the server. Verify that SQL Server has enough CPU allocated to it. Prioritize the other applications running on the server and decide if any of the other processes can be migrated to another server, thereby freeing CPU resources to the SQL Server.

Transaction Log Bottleneck

Some of the databases suffer from transaction log bottleneck. Log wait consumes x% of the In MS-SQL time.

Table 13-31   Transaction Log Bottleneck findings

Column	Description
What to do next	■ Examine which database in the instance suffers from transaction log bottleneck, in the Activity tab. ■ Examine the In MS-SQL breakdown of the applications, in the Activity tab. ■ Examine the load on the devices holding the transaction logs, in the Statistics tab. ■ Examine Log Flush counters, in the Statistics tab.

Table 13-31 Transaction Log Bottleneck findings

Column	Description
Advice	The transaction log is the file that experiences many writes and few reads. Every update to the data is eventually written to the transaction log. Since every database has its own transaction log, the first thing to do is to try to pinpoint the specific database.
	If all of databases are experiencing a transaction log bottleneck, try examining the recovery interval (min) configuration.
	When you have identified which database is experiencing a transaction log bottleneck, examine the applications using this database.
	A bottleneck in the transaction log indicates that your application performs commits at a very high or very low frequency. This means that many records must be saved at the same time. The challenge is finding how many commits to do and how many records to save in each commit.
	You can also consider moving the transaction logs to different and faster devices.

Extensive Internal Wait

The SQL Server instance has spent much of its time waiting for Internal Wait.

Table 13-32 Extensive Internal Wait findings

Column	Description
What to do next	■ Examine which database in the instance suffers from Internal Waits, in the Activity tab. ■ Examine the Internal Wait category using the Internal Waits view, in the Activity tab. ■ Examine a specific Internal Wait in the Statistics tab (SQL Server 2005 only).

Table 13-32 Extensive Internal Wait findings

Column	Description
Advice	An Internal Wait is divided into the following substates: Buffer Pool Groups together the events that show contention on pages in the buffer pool. Buffer wait on tempdb pages are considered to be part of the tempdb wait. Latch The session is waiting for an internal lock to be released. Parallel The session is waiting for one of its sub-threads to complete its operation. DTC Aggregates waits that occur when Distributed Transaction Coordinator (DTC) sessions have to wait for each other. This state is only available for SQL Server 2005 instances. DB Mirror 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. This state is only available for SQL Server 2005 instances. Profiler Aggregates a number of states associated with the SQL Profiler and lets you see how much of the database resources it consumes. This state is only available for SQL Server 2005 instances. Memory Aggregates several types of waits that indicate that a session is waiting for memory to be allocated to it. This state is only available for SQL Server 2005 instances. Backup Includes the wait type that commonly occurs when a Backup command is performed. This state is only available for SQL Server 2005 instances. Other Internal waits Aggregates the following types of waits: Full text waits Includes wait types dedicated to the full text indexing service and appears whenever a full text index is in progress. HTTP waits Includes waits that occur when HTTP and SOAP operations are executing. Query notifications Aggregates a number of states associated with the synchronization of Query Notification sessions. This state is only available for SQL Server 2005 instances.

Tempdb Major Growth

The tempdb database has experienced major growth. The tempdb database is used for temporary storage for sorting, joining, and, in SQL Server 2005, for row versioning.

Table 13-33   Tempdb Major Growth findings

Column	Description
What to do next	Perform one of the following options: ■ Examine the current tempdb content to find out which resource type is being consumed the most: User temporary tables Internal objects such as temporary system tables caused by Sort, Hash Join or Spooling operations. ■ Examine the type of statements heavily using the tempdb resource in the Statements tab. ■ Examine the Statistics to find out the number of files/tables creation number and rate. ■ Examine the Tempdb Usage to find out open transactions preventing tempdb to shrink.
Advice	■ Examine the type of statements heavily using the tempdb resource in the Statements tab. Tune sorts Try reducing the space or memory required for Sort operations by reducing the number of sorted columns or by filtering the rows to be sorted. Hash and Merge joins are temporary storage consumers. Try to find a way to eliminate the sort by using pre-sorted information like creating an index on the sorted columns. Check the values of the min memory (MB) and index create memory (KB) instance parameters. Use the index create memory option to control the amount of memory used by index creation sorts. The index create memory option is self-configuring and should work in most cases without requiring adjustment. However, if you experience difficulties creating indexes, consider increasing the value of this option from its run value. Query sorts are controlled through the min memory per query option. ■ Tune Hash Joins: Verify that there are **no missing** indexes Verify that there are **adequate** indexes Search for a missing **WHERE** clause Search for a **non-sargeable** expression. A **non-sargeable** expression is an expression preventing the optimizer of using the index in the ideal way or not at all like a function expression. ■ Tune Index Spool, Table Spool and Row Count Spool: Spooling is internally temporary tables created by SQL Server on the tempdb database. Spooling may cause tempdb overhead and thus resulting in instance performance degradation caused by tempdb wait or tempdb major growth. Try to eliminate spooling operations by rephrasing the query. Try to reduce the number of logical reads or writes. ■ Tune row versioning: If a version store is not shrinking, it implies that a long-running transaction is preventing version store cleanup. ■ Examine tempdb usage when the applications explicitly create temporary tables. Try creating indexes on those tables after they are populated. ■ Examine the Tempdb Usage to find out open transactions preventing tempdb to shrink.

High CPU Wait

The instance used SQL Server resources while **waiting for CPU**.

Table 13-34   High CPU Wait

Column	Description
What to do next	■ Examine the high CPU usage statements. ■ Examine the CPU usage to find the number of waiting for CPU tasks. ■ Examine where the schedulers are not evenly loaded on the instance. ■ Examine the overall CPU usage (using Insight OS) to find out if other processes occupy the CPUs.

About Storage findings

The following Storage findings can help you tune your system:

- [Storage Contention on Device (Clariion)](#)

- [Storage Contention on Device (Symmetrix Thick)](#)

- [Storage Contention on Device (Symmetrix Thin)](#)

- [Storage Contention on Device (Symmetrix F.A.S.T. VP)](#)

- [Storage Contention between Log and Datafiles](#)

- [Unbalanced Storage Devices Activity](#)

- [Storage with Low Cache Hit Ratio](#)

- [Storage Contention on tempdb](#)

Storage Contention on Device (Clariion)

The instance spent **X%** of its resources waiting for **I/O on the specified storage device.**

The fact that a storage device (LUN) is causing a lot of I/O waits could be caused from an intensive load or as a result of two sorts of contentions: a logical contention (e.g. imbalanced activity of the database) or a physical contention (e.g. one of the underlying physical devices is being shared with another heavy I/O consuming activity).

Table 13-35   Storage Contention on Device findings

Column	Description
What to do next	■ Examine the <u>deviceactivity</u> over time and database files contention. ■ Examine storage contention in <u>PhysicalDisks</u>, <u>Devices</u> and <u>Files</u> on the same raid group. ■ Examine <u>disksstatistics</u>.
Advice	■ If the device is loaded by the monitored database only and by a singular entity (e.g. a file, object, or partition), consider splitting this load (e.g. separating the objects in the file, partitioning the object, etc). ■ To relieve inter application logical contention, check if the database's I/O activity is balanced. Spread heavy I/O consuming files across the storage devices, to avoid a situation in which few heavy files reside on the same storage device. ■ To relieve intra application logical contention, check whether there are additional applications using the storage device. For example, if the number of I/O requests processed by the storage device is significantly higher that the requests sent by the database, it means that the storage device is being used by an additional application. ■ To relieve physical contention, check whether there is significant I/O activity in the underlying shared physical disks and raid group. Another potential cause of contention are the EMC adapters (front director and disk director). If the load is imbalanced, consult with the storage administrator about relocating the information to other disks which reside on a more vacant location. ■ Consider storage tiering - a faster device may reduce the I/O wait time significantly.

Storage Contention on Device (Symmetrix Thick)

The instance spent **X%** of its resources waiting for **I/O on the specified storage device.**

The fact that a storage device (LUN) is causing a lot of I/O waits could be caused from an intensive load or as a result of two sorts of contentions: a logical contention (e.g. imbalanced activity of the database) or a physical contention (e.g. one of the underlying physical devices is being shared with another heavy I/O consuming activity).

Table 13-36 Storage Contention on Device findings

Column	Description
What to do next	■ Examine the <u>deviceactivity</u> over time and database files contention. ■ Examine storage contention in <u>PhysicalDisks</u>, <u>Devices</u> and <u>Files</u> on the same raid group. ■ Examine <u>disksstatistics</u>.
Advice	■ If the device is loaded by the monitored database only and by a singular entity (e.g. a file, object, or partition), consider splitting this load (e.g. separating the objects in the file, partitioning the object, etc). ■ To relieve inter application logical contention, check if the database's I/O activity is balanced. Spread heavy I/O consuming files across the storage devices, to avoid a situation in which few heavy files reside on the same storage device. ■ To relieve intra application logical contention, check whether there are additional applications using the storage device. For example, if the number of I/O requests processed by the storage device is significantly higher that the requests sent by the database, it means that the storage device is being used by an additional application. ■ To relieve physical contention, check whether there is significant I/O activity in the underlying shared physical disks and raid group. Another potential cause of contention are the EMC adapters (front director and disk director). If the load is imbalanced, consult with the storage administrator about relocating the information to other disks which reside on a more vacant location. ■ Consider storage tiering - a faster device may reduce the I/O wait time significantly.

Storage Contention on Device (Symmetrix Thin)

The instance spent **X%** of its resources waiting for **I/O on the specified storage device.**

The fact that a storage device (LUN) is causing a lot of I/O waits could be caused from an intensive load or as a result of two sorts of contentions: a logical contention (e.g. imbalanced activity of the database) or a physical contention (e.g. one of the underlying physical devices is being shared with another heavy I/O consuming activity).

Table 13-37 Storage Contention on Device findings

Column	Description
What to do next	■ Examine the <u>deviceactivity</u> over time and database files contention. ■ Examine storage contention in <u>PhysicalDisks</u>, <u>Devices</u> and <u>Files</u> on the same raid group. ■ Examine <u>disksstatistics</u>.

Table 13-37 Storage Contention on Device findings

Column	Description
Advice	■ If the device is loaded by the monitored database only and by a singular entity (e.g. a file, object, or partition), consider splitting this load (e.g. separating the objects in the file, partitioning the object, etc). ■ To relieve inter application logical contention, check if the database's I/O activity is balanced. Spread heavy I/O consuming files across the storage devices, to avoid a situation in which few heavy files reside on the same storage device. ■ To relieve intra application logical contention, check whether there are additional applications using the storage device. For example, if the number of I/O requests processed by the storage device is significantly higher that the requests sent by the database, it means that the storage device is being used by an additional application. ■ To relieve physical contention, check whether there is significant I/O activity in the underlying shared physical disks and raid group. Another potential cause of contention are the EMC adapters (front director and disk director). If the load is imbalanced, consult with the storage administrator about relocating the information to other disks which reside on a more vacant location. ■ Consider storage tiering - a faster device may reduce the I/O wait time significantly.

Storage Contention on Device (Symmetrix F.A.S.T. VP)

The instance spent **X%** of its resources waiting for **I/O on the specified storage device.**

The fact that a storage device (LUN) is causing a lot of I/O waits could be caused from an intensive load or as a result of two sorts of contentions: a logical contention (e.g. imbalanced activity of the database) or a physical contention (e.g. one of the underlying physical devices is being shared with another heavy I/O consuming activity).

Table 13-38 Storage Contention on Device findings

Column	Description
What to do next	■ Examine the <u>deviceactivity</u> over time and database files contention. ■ Examine storage contention in <u>PhysicalDisks</u>, <u>Devices</u> and <u>Files</u> on the same raid group. ■ Examine <u>disksstatistics</u>.
Advice	■ If the device is loaded by the monitored database only and by a singular entity (e.g. a file, object, or partition), consider splitting this load (e.g. separating the objects in the file, partitioning the object, etc). ■ To relieve inter application logical contention, check if the database's I/O activity is balanced. Spread heavy I/O consuming files across the storage devices, to avoid a situation in which few heavy files reside on the same storage device. ■ To relieve intra application logical contention, check whether there are additional applications using the storage device. For example, if the number of I/O requests processed by the storage device is significantly higher that the requests sent by the database, it means that the storage device is being used by an additional application. ■ To relieve physical contention, check whether there is significant I/O activity in the underlying shared physical disks and raid group. Another potential cause of contention are the EMC adapters (front director and disk director). If the load is imbalanced, consult with the storage administrator about relocating the information to other disks which reside on a more vacant location. ■ Consider storage tiering - a faster device may reduce the I/O wait time significantly.

Storage Contention between Log and Datafiles

The instance spent **X%** of its resources waiting for **I/O on the specified storage device**.

The device contains both Log files and Datafiles.

Transaction Log files are frequently accessed by the database. The majority of the operations performed are writing commands, which cause a heavy load on the underlying disks.

As these files are considered heavy I/O consumers, it is highly recommended to place them on a separate disk without other database files. Separating the Transaction Log files by placing them on different volumes (e.g. E:/ and F:/) may not be enough, as the storage devices (LUNs) and physical disks may be shared between several file systems and volumes.

Table 13-39   Storage Contention between Log and Datafiles findings

Column	Description
What to do next	■ Examine the <u>deviceactivity</u> over time and database files contention. ■ Examine storage contention in <u>PhysicalDisks</u>, <u>Devices</u> and <u>Files</u> on the same raid group. ■ Examine <u>disksstatistics</u>.
Advice	It has been detected that the Transaction Log files share the storage devices (LUNs) with other database files. Consult the storage administrator about provisioning the storage devices (LUNs) better to avoid this.

Unbalanced Storage Devices Activity

The instance I/O activity is **not balanced** across storage devices.

There are several storage devices (LUNs) allocated to the instance. However, the I/O activity is not spread evenly across these storage devices. The contention on the heavy storage devices increases the response time for the activities run on them. Such a situation can be caused by imbalanced internal database activity, contention on the storage device by other applications or an inefficient RAID policy.

Table 13-40   Unbalanced Storage Device Activity findings

Column	Description
What to do next	■ Compare the <u>storagedevicesactivity</u> over time. ■ Examine the <u>storagedevicesstatistics</u>.
Advice	■ In the activity tab, check which database files are the most I/O consuming and spread them evenly across the storage devices. ■ Consult with the storage administrator and check for other applications using the same storage devices or their underlying physical disks. ■ Consult with the storage administrator about the RAID policy. A different striping may spread the I/O load across the storage devices.

Storage with Low Cache Hit Ratio

The instance was waiting for I/O on storage devices with low cache read hit ratio.

I/O requests sent to the EMC storage array are processed on the storage internal cache. EMC runs a "prefetch" mechanism that predicts the future blocks to be requested and load them to the cache. Requests that are not being served by the cache access the underlying disks, and are significantly slower. The cache can be shared by different storage devices (LUNs), and may potentially become a layer of contention between I/O activities which are being served by those storage devices.

Table 13-41   Storage with Low Cache Hit Ratio findings

Column	Description
What to do next	Examine the <u>storagedevicesstatistics</u> and hit radio.
Advice	It has been detected that the storage devices (LUNs) that serve the instance get bad cache performance (i.e. low "hit ratio"). To relief the cache contention, consult the storage administrator about the following: ■ Expanding the storage cache allocated to the database devices. ■ Enabling "EMC Cache Partitioning", to isolate the instance cache and avoid external contentions.

Storage Contention on tempdb

Tempdb datafiles are frequently accessed by the instance. The majority of the operations performed are writing commands, which cause a heavy load on the underlying disks.

As these files are considered heavy I/O consumers, it is highly recommended to place them on a separate disk without other database files. Separating the tempdb datafiles by placing them on different volumes (e.g. E:/ and F:/) may not be enough, as the storage devices (LUNs) and physical disks may be shared between several file systems and volumes.

Table 13-42   Storage Contention on tempdb findings

Column	Description
What to do next	■ Examine the <u>deviceactivity</u> over time and database files contention. ■ Examine storage contention in <u>PhysicalDisks</u>, <u>Devices</u> and <u>Files</u> on the same raid group. ■ Examine <u>disksstatistics</u>.
Advice	It has been detected that the undo tablespace files share the storage devices (LUNs) with other database files. Consult the storage administrator about provisioning the storage devices (LUNs) better to avoid this.

Glosary

abandonment rate
In Web, a counter that keeps track of the percentage of users that abandon the loading of the Web page before it completes downloading.

action
An operation that Alerts FocalPoint automatically performs when detecting a warning or critical status for a specific metric. According to the defined action, Alerts FocalPoint opens a message box, sends an email or SNMP trap, or executes a program.

action item (manual)
Some installation steps cannot be executed automatically by Precise Installer. The administrator is required to execute them manually. Action Items are presented (when necessary) at end of installation or product update.

AdminPoint
The central administration console of Precise that facilitates the maintenance, configuration, and management of all installed Precise components, such as monitoring the status of all Precise agents and PMDB processes, getting license information, starting and stopping the agents, getting log data on agents and events, changing PMDB settings, and installing patches.

See also PMDB.

See also agent.

advice
In Oracle, an algorithm that is designed to recommend on gathering statistics, creating new indexes, change existing indexes, and add or delete hints to make Oracle's Optimizer choose a better access path and make the statement perform better. Can be activated from any DML (Data Manipulation Language) statement.

agent
A program that runs on a server machine to collect, process, or communicate performance information. The Precise installation consists of multiple agents.

alert
The state of an Alerts metric that has reached a near-critical or critical status. An alert is issued by Alerts, triggering an action and informing of a problem that has occurred or is likely to occur within the area sampled by the specific metric.

See also action.

Alerts
These product provide alerts to problematic conditions before they turn into performance problems, based on predefined metrics and thresholds. Alerts can automatically perform an action, such as displaying a pop-up message, sending an email message or SNMP trap, or running a program.

Alerts FocalPoint
An agent that receives data from the InformPoint agents, stores it, and performs any action that has been user-defined for that specific alert, such as displaying a pop-up message, sending an email message or SNMP trap, or running a program.

See also InformPoint.

alert type
In Alerts, the status of all metrics belonging to a metric group or a monitored instance, indicating the current performance level through colors.

See also alert.

See also metric.

application server metric
In J2EE, a metrics that is provided by the application server or by customer code. This can include metrics published by the Java Management Extension (JMX) APIs or vendor-specific APIs, such as IBM's Performance Management Interface (PMI).

AppTier
The abbreviation for an application tier in a Precise environment. An AppTier contains one or more instances of the same technology and purpose. Application tiers do not necessarily correspond to distinct physical servers: in many cases, the tiers are logical, with one server running multiple AppTiers or one AppTier spans a cluster of servers. A Precise environment may contain multiple AppTiers on the same technology. For example, you may group J2EE instances (JVMs) into a J2EE Presentation AppTier and a J2EE Business Logic AppTier. Segmenting application service time into the contribution of individual application tiers is helpful in identifying the source of performance problems. Analyzing the performance and behavior of each tier separately is crucial for isolating the root causes of performance problems.

cabinet	In Oracle, SQL Server, DB2, and Sybase, the highest logical level in the SQL workspace hierarchy. A cabinet contains folders and, within folders, statements
call path	A subset of a stack trace including only those methods that have been instrumented.
client cache	In Web, a counter that keeps track of the percentage of requests taken from the client cache (http status 304).
client-side collection	See Web client browser-side.
cluster	A group of servers or instances that are configured to be treated as a single entity. In Oracle, all Oracle instances of the same Oracle database (RAC configuration) are treated as a cluster and presented in the Precise for Oracle UI as a Database. In J2EE, a group of JVMs that serve the same application can be defined as Cluster (even if they are not defined as a cluster in the application server configuration). All JVMs of the same cluster share the same configuration.
collapsed access plan	In SQL Server and Sybase, the access plan of a unique group of statements or batches that belong to the same collapsed statement or batch but have different access plans. This can differ due to the constants in the text of the original (not collapsed) statements.
	See also explained statement.
	See also collapsed statement.
collapsed statement	In Oracle, SQL Server, Sybase, and DB2, a statement whose constants are replaced with parameters. Each collapsed statement can have several access plans, according to the occurrences of its statements.
	See also collapsed access plan.
Collector	The program that runs on the monitored server to collect performance information.
	Some technologies allow a single Collector (agent) to serve multiple instances running on the same server. Other technologies require a dedicated Collector per monitored instance.
	In most cases, it must be installed on the monitored server. Collectors for SAP, SQL Server, and Sybase may reside on a remote server.
	See also agent.
collector slice size	In J2EE, the smallest unit of data presentation and storage. It can be configured as 30 seconds or five minutes. If SmartLink is configured, relevant Web and Microsoft .NET Collectors will report to SmartLink using the same collector slice size. Regardless of the defined collector slice size, the most granular slice size presented on both the Precise for Web UI and the Precise for Microsoft .NET UI will be 15 minute intervals.
completion rate	In Web, a counter that keeps track of the percentage of users that completed the loading of a Web page.
count	The number of occurrences observed during a measurement interval.
CPU time	The average amount of time consumed by the operating system actively processing instructions on behalf of a running activity.
critical status	In Alerts, the status represented by a red bullet indicating that the value of the sampled metric has exceeded the near-critical and critical threshold values.
	See also metric.
Cross-AppTiers	In Insight, Report Manager, and Alerts, a perspective that provides high-level information about the performance of all AppTiers in an environment, including operating system data. It provides a holistic view of the entire environment and helps understand how the AppTiers interact.
	See also AppTier.
current data	In SmartLink, the end-to-end activity, detected by Precise over the last five minutes, including activities that are still in progress. In all databases products, a workspace that presents Current Sessions and the SQL statements that were executed including statements that are still in progress.
custom	Represents invocations of a type not specifically defined.
	See also work time.
customized metric	In Alerts, a user-defined metric that measures site-specific parameters.
drilldown	Within Precise products and Insight SmartLink, the filtering of analyzed data by clicking a specific entity. Then, additional information about the selected entity is presented, plus a breakdown of its activity by another entity. For example, drilldown on a transaction and get an overtime graph of the selected transaction, plus a list of top users that executed the selected transaction.
entry point	Usually a top-level HTTP or EJB request. An entry point can originate when a user clicks in a browser or an E-commerce server invokes a remote EJB. J2EE makes a distinction between service request invocations and other invocations. The first HTTP or EJB invocation within a call path is designated as an entry point.

environment	The highest Precise logical group. It may contain multiple AppTiers of various technologies that serve an application together. For example, a Payroll Production environment may contain all Web servers, application servers, transaction managers, databases, and servers that serve this application. Alternatively, it may contain any set of instances that form together an administrative group. Since Precise version 8.5, an instance can belong to more than one environment. Let's say a single Oracle database serves two different applications plus the DBA wants to associate this database with a group of other databases under his responsibility regardless of served applications. In this case the Oracle database will be associated to three different Precise environments.
	See also AppTier.
	See also instance.
ERP Extension	In Oracle, DB2, and SQL Server, an extension to the Collector that provides detailed information on the activities and resource consumption of packaged application components. It correlates the database information, and the packaged application information and lets you see users, transactions, reports, and other elements of ERP applications, such as Oracle Applications, SAP, PeopleSoft, and Siebel.
event	In Alerts, the occurrence of a sampling or progress. A sampling occurrence occurs every time a metric samples. A progress occurrence occurs when a metric's progress status is changed or when the investigated status reaches the end of the given investigation time. In AdminPoint, all occurrences reported by Precise agents, including informational events, warnings, and errors related to one of the agents. All events are shown in the Events view.
	See also metric.
	See also progress.
executions	In Oracle, SQL Server, DB2, and Sybase, the number of times a SQL statement was executed during the selected time frame. In SmartLink and SAP, the number of times a transaction was executed during the selected time frame.
explained statement	In Oracle, SQL Server, DB2, and Sybase, a statement whose access path (chosen by the RDBMS Optimizer) is clarified and translated into a visual display. Explained results include information on the objects referenced by the statement and the operations performed on these objects.
extended collection	In Oracle, a function that proactively specifies a future period during which Oracle activity data is collected and organized for subsequent analysis. Extended collections are an easy means to view collected information, assess application resource consumption, and identify bottlenecks that are inhibiting application performance and end-user productivity.
Federated Precise	Federated Precise is a version of Precise (starting with version 8.5) that can manage multiple Precise installations within unified StartPoint and AdminPoint screens, displaying and managing all environments, instances, and installations.
findings	A ranked list of top performance problems in the selected context. They are presented in Oracle, SQL Server, .NET, and J2EE. Each finding provides: a short explanation of the problem, detailed background information, and most important: one or more links to further investigate the problem and possibly get more granular findings on the selected context.
first byte time	In Web, a counter that keeps track of the time that it takes from the moment a new Web page is called until the first byte arrives back from the Web server.
FocalPoint	An agent that communicates with the Listeners installed on the monitored servers, receives data from the Collectors, periodically processes and stores this data in the PMDB, and serves UI requests.
folder	In SQL Server, Oracle, DB2, and Sybase, the intermediate logical level in the SQL workspace hierarchy. Folders are grouped into cabinets and contain SQL statements.
	See also cabinet.
Framework Installer	The application that facilitates the installation of Precise framework components. It can be invoked from the installation DVD to install a new Precise deployment. It can also be invoked from an existing Precise deployment to install an additional framework node and attach it to the originating Precise deployment.
	See also Framework node.
Framework node	A set of FocalPoint agents that are installed together and manages a set of monitored instances. The performance data of these instances will be loaded into a dedicated PMDB. A single Precise deployment may contain multiple framework nodes (using a separate PMDB for each node). An environment cannot span over multiple framework nodes.

garbage collection	An automatic process in the Java runtime environment that periodically reclaims memory used by objects that are no longer referenced. The process can impact an application's performance while memory is being reclaimed. Java programmers may initiate garbage collection explicitly.
grouper	In Web, the identifier that is used to group other identifiers, such as sites or URLs.
hint	In Oracle, an instruction directed at the Oracle Optimizer that includes considerations for an execution plan. The Oracle Optimizer will build an execution plan based on the hint, ignoring its own set of considerations.
hour group	A unit that reflects the type and level of activity within the system at different times. By defining the times of the day that are peak and off-peak, or day and night, the performance analysis can be focused on those particular times of the day. If, for example, most performance problems occur within nighttime and weekend batches, it can be useful to focus only on them.
InformPoint	In Alerts, an agent that retrieves performance data from all installed Precise products, analyzes it, and sends an alert to the Alerts FocalPoint when the predefined thresholds are exceeded. See also agent.
Insight	The Precise product family that facilitates the process of monitoring and correlating system performance. It consists of Insight. See also Report Manager. See also FocalPoint.
Insight FocalPoint	An agent that receives performance information from Insight Savvies, which monitor the environment. Insight FocalPoint then correlates, processes, and stores this information in a centralized location. The Insight performance history is stored in the PMDB. See also Savvy. See also PMDB.
instance	A monitored object of a specific technology. The following list specifies what constitutes an instance for the various supported technologies: J2EE - a Java Virtual Machine (JVM - a logical name set by the user), Microsoft .NET - a Common Language Runtime (CLR - a logical name set by the user), Oracle Applications - an Oracle Applications Form server, Oracle - an Oracle instance, SAP - a SAP system, SQL Server - an SQL instance, Sybase - a Sybase instance, Tuxedo - a Tuxedo domain, Web - a Web server, WebSphere MQ - an IBM WebSphere Queue Manager, DB2 - a DB2 non-partitioned database or a DB2 database partition. During installation, the instance is associated with one AppTier and environment. An Instance can be moved to a different environment or associated with multiple environments without re-installation nor losing historical data. See also environment.
instance statistics	See performance counter.
instance/database changes	In SQL Server and Oracle, the part of the Collect Schema Changes process that captures instance definition changes and database option changes and saves them in the PMDB.
instrumentation	The process of inserting fault-tolerant recording hooks in Java byte code, .NET MSIL, HTML pages, or other monitored components, resulting in the capture of performance metrics. In J2EE, a mechanism that enables collecting performance information when an application is executed. The process involves inserting special fault-tolerant recording hooks into application class objects. In WEB, the insertion of recording hooks into HTML pages can either be in memory (Dynamic Instrumentation) or file-based (Static Instrumentation).
instrumentation context	See invocation context.
internal invocation	The process of invoking a request from an HTTP request (Servlet or JSP) or an EJB. J2EE displays an internal invocation like any other invocation.
invocation	An execution of a J2EE entity (a Servlet, EJB, SQL Statement, method, and so on). When mentioned in plural (Invocations), means the amount of times that the entity was executed.
invocation context	The context within which a method is invoked. For example, if Method A is invoked both from Method B and from Method C, there will be two different invocation contexts for the performance metrics collected for Method A, one for when it is invoked from Method B and one for when it is invoked from Method C. See also call path and instrumentation context.
JRE (Java Runtime Environment)	As the runtime part of the Java software, the combination of the components that enable the execution of a Java program: a Java virtual machine, the core class libraries, and the files that form the Java platform.

JSP (Java Server Page)	An HTML page with special tags for Java scripting. An application server processes the tags and generates a Servlet.
JVM (Java Virtual Machine)	An instance of a JRE that executes Java programs. A server-side Java application server is itself a Java program that runs inside a JVM. Servlets, JSPs and EJBs are Java programs (applications) that run within the application server's JVM. J2EE monitors the JVM running the application server and the server-side Java applications within the application server.
JVM Heap Memory	The amount of real computer memory that is allocated to the JVM for executing Java programs.
key metric	In Alerts, a parameter that monitors a very important performance aspect. The status of the Oracle instance (up/down), for example, is crucial for system performance. If the instance is down, this is the first problem that needs to be solved. Marking a metric as key metric ensures that a critical alert raised for this metric receives top priority by the person that is responsible to handle alerts at any time. It can also help determine which alert to handle first in case of multiple alerts. In Alerts, key metrics are always displayed at the top of a metric table when it is sorted by alerts so that they get immediate attention.
Listener	The agent that facilitates the communication between the various Precise agents across different servers must be installed on every server where Collectors or FocalPoint agents are installed. The Listener allows communication with all other agents installed on the monitored server, while only the Precise Listener port is known by other servers.
login name	In SQL Server and Sybase, the session identifier that represents the credential used to connect to the database. When an ERP extension is installed, the user name of the packaged application's client overrides the login name. For example, when SAP extension is installed, the SAP user name overrides the login name.
machine	A session identifier. A machine as sampled, for example, by the SQL Server Collector is the identification of the machine where the client process executes. Machine is also sampled by Oracle, DB2 and Sybase. In Insight terminology Machine is called Client machine.
Main framework node	A main framework node is the single point for login and also serves as the Precise FocalPoint for the entire deployment.
major collection count	Number of estimated major garbage collection events that occurred during the last J2EE collector slice. A major garbage collection can stop the application while JVM heap memory is being reclaimed. See also garbage collection.
major garbage collection time	Percentage of time spent by the JVM executing major garbage collection events during the displayed interval. See also garbage collection.
MBean	A Java object that represents a manageable resource. In J2EE, MBeans, or Managed Beans, are used for application server metrics.
memory logger interval	The interval at which the J2EE Collector gathers JVM heap memory data. All snapshots of the memory logger's data collected according to the memory logger interval are summarized with counters in the current aggregation interval. The memory logger interval's time span is typically a small fraction of the aggregation interval.
metric	In Alerts, a query that helps measure performance in the environment. Three types of metrics are available: System metrics relate to the internal resources, operations, and objects of the monitored infrastructure; application metrics reflect the way the applications perform; user-defined metrics can be customized to specifically relate to a site. When a metric's value exceeds one of the defined thresholds, its status changes to near-critical (yellow bullet) or critical (red bullet). See also key metric.
metric set	In Alerts, a unit that groups metrics that measure related performance aspects. The following metric sets exist: Status: includes metrics that alert to functional problems related to the instance. Performance: includes metrics that alert to performance problems related to the instance. Load: includes metrics that alert to instance-related load problems that may later cause errors or crashes. Service: includes metrics that alert to instance-related SLA breaches. The metrics in this set are sampled by Insight. Performance Trending: includes metrics that alert to potential future performance problems. The metrics in this set are sampled by Report Manager.

	Load Trending: includes metrics that alert to potential future load problems. The metrics in this set are sampled by Report Manager. Customized: includes user-defined metrics. Precise Status: includes metrics that alert to the near-critical or critical status of the installed Precise environment.
module	A session identifier. In Oracle it contains the value of MODULE column in V$SESSION table. In DB2 it contains the command name of non-SQL statements or the package name for SQL statements.
near-critical status	In Alerts, a status indicating that the value of the sampled metric has exceeded the defined near-critical threshold. A near-critical status is indicated by a yellow bullet.
network time	In Web, a counter that keeps track of the time spent on network activity from the server side perspective. This includes the time to read the request from the network and the network time to send the response back to the client.
node	See Framework node. Note: when creating a new installation from DVD, it would be called "Framework." In the UI screens (columns in tables, choosing a system for a new environment etc), we would call them "Nodes." For example: choose a node for the new environment. When adding a new system within AdminPoint, it would be called a "Framework node".
packaged application	An application that is created and/or maintained by a third party and is not custom-built to one's specific needs. The following packaged applications have special treatment by Precise: SAP, Oracle Applications, PeopleSoft, and Siebel. They are harder to modify because the application code is either not available or hard to understand. See also ERP Extension.
page size	In Web, a counter that keeps track of the amount of data loaded from the server to display the page.
page views	In Web, a counter that keeps track of the number of Web pages viewed at a specific Web site during a selected time period.
parent metric	In Alerts, a joining of several child metrics. Each time a parent metric samples, it gathers data from a set of child metrics and presents it as a single metric query. The individual child metric values are displayed on the Thresholds tab of the Properties dialog box in Alerts. See also submetric.
performance counter	In SQL Server, a Windows performance counter as reported by the operating system. In Oracle, an instance level statistic as gathered from some Oracle V$ views. In Sybase, a sysmonitors counter collected by Sybase. In Web, an operating system or a Web server performance counter.
PMDB	The Precise data warehouse of performance and availability data. It can be hosted on an Oracle or SQL Server database.
portal server	An application server for Web-based applications that commonly provide personalization, single sign-on, and content aggregation from different sources and that host the presentation layer of Information Systems. Aggregation is the action of integrating content from different sources within a Web page. A portal may have sophisticated personalization features to provide customized content to users. Portal pages may have a different set of portlets creating content for different users.
portlet	A Java-based Web component, managed by a portlet container, that processes requests and generates dynamic content. Portlets are used by portals as pluggable user interface components that provide a presentation layer to Information Systems.
Precise deployment	An independent Precise system. It contains and manages various agent types and provides centralized monitoring and administration. A Precise deployment may contain multiple framework nodes. One of them is defined as the main framework node and it manages all other nodes of the Precise deployment.
Precise FocalPoint	See Precise FocalPoint agent.
Precise FocalPoint agent	An agent that manages all agents in a single Precise deployment. Additional product FocalPoint agents manage specific technologies and resources.
Precise Proxy agent	When multiple Framework Nodes are managed by a single Precise deployment, the Precise FocalPoint agent manages all of the agents of the main Framework node, while every other framework node has a Precise Proxy agent to manage all its agents.
Precise_root	This is the term used in a path for the Precise installation directory. The terms *<i3_root>* or i3 root can appear in text too.

program	A session identifier in Oracle, DB2, SQL Server, Sybase, Insight, and Alerts. A program as sampled, for example, by the SQL Server Collector, is the name of the executable that connects to the database. Applications that do not set the application name have N/A as program. When a SAP extension is installed, the SAP transaction overrides the program, and it may change during the application's lifetime. When a Siebel extension is installed, the Siebel views override the program, and it may change during the application's lifetime. In Insight, program is the name of an executable as recognized by the operating system. If an executable is invoked from a script (a batch or a shell), the script is displayed as the command entity. In Alerts, program is part of the customized metrics definition and is the name of the executable or stored procedure executed in the database that runs when the metric is sampled. In action definitions, program is the name of the executable that will run if the metric exceeds its threshold.
progress	In Alerts, the management state of a metric for which an alert has been raised. The following statuses exist:
	Open: An alert is raised.
	Investigated: The alert is taken care of.
	Closed: The problem has been solved.
Proxy FocalPoint	See Precise Proxy agent
recommended index	In SQL Server, a function that uses the Microsoft Index Tuning Wizard to recommend on adding indexes or statistics for a selected statement, batch, or table. Based on the results of this function, the Optimizer will choose a better access plan and make the respective statement or batch perform better. For statements and batches, recommendations are based on the content of the statement or batch. For tables, recommendations are based on all the statements that are stored in the PMDB, are executed during the selected time period, and have an average duration time greater than the value defined in the registry (where 0 is the default).
	See also advice.
related SQL	In Oracle, a generated statement that uses alternative syntax to access the database in different ways and returns the same output as the original statement.
relative frequency	Number of contributor invocations per entry point. For example, if a service request calls three methods each time it is invoked, the Relative Frequency for the method is three invocations per service request execution. Similarly, if a service request calls one method every other time it is invoked (half of the time), the Relative Frequency for the method is 0.5 invocations per service request execution.
rendering time	In Web, a counter that keeps track of the time that it takes for a Web page to be loaded from the moment the first byte arrives until the Web page is fully loaded or the user interrupts or abandons the loading process.
report	A collection of queries, programming code, and layout settings that Report Manager executes to generate graphical results like tables and charts.
Report Manager	Uses historical information to identify problematic conditions, track long-term performance, volume trends and patterns, view availability problems over long periods of time and on different levels, compare the performance of similar systems, correlate between performance metrics of different products, assist in capacity planning, and generate demand-driven, user-defined reports.
Report Manager FocalPoint	An agent that examines the PMDB tables to produce scheduled performance reports.
	See also PMDB.
report parameter	In Report Manager, a keyword used in a report. Its value is set during the report execution. A parameter's value can be updated either permanently or for the current execution only.
report property	In Report Manager, the attributes that define a specific report, consisting of report parameters and scheduling information.
	See also report parameter.
request	In Web, a counter that keeps track of the number of HTTP requests sent for a viewed entity. **request**
error	In Web, a counter that keeps track of the percentage of requests completed with an HTTP error.
sampling	In Alerts, the process during which a metric queries a Precise product for a specific instance, retrieves values, and calculates the metric's alert level.
sampling base	In Alerts, the start time of a scheduled sampling process (by default Sunday, 00:00 AM).
sampling period	In specific Alerts metrics, the time frame during which statistical data is returned from other Precise products. Such a metric is for example Oracle's General Behavior metric, which returns database behavior for a certain period of time.

sampling rate	In Alerts, the frequency of a metric's regular sampling schedule. The sampling rate is measured in minutes.
Savvy	An Insight agent that collects AppTier-specific performance data.
scalability	A system's ability to withstand load. For example, positive scalability means that the system continues to function properly even when it is called upon to service a larger number of users.
schema changes	In Oracle and SQL Server, a process that captures schema changes and saves them in the PMDB. In SQL Server, instance configuration parameters and database option changes are also captured.
Script Installer	A component of Precise Agent Installer that enables adding a small JavaScript script, also known as a Web browser-side agent, to the Web pages of your Web site.
server	Also called a host machine or server machine. The combination of a computer and associated operating system software that is accessed by multiple clients to perform requested functions and typically returns results to the clients.
server-side collection	In Web, the collecting of performance data from the Web server instance through the Web server agent.
service + network time	In Web, a counter that keeps track of the service and network time, including the total amount of time the request took to reach the server.
Service Level Agreement	See SLA (Service Level Agreement).
service time	In Web and SmartLink, the time elapsed from when a request is received by the server to the time a response is sent back to the computer that made that request. The service time is measured on the server side. In J2EE, the time it takes an invocation to complete execution. In other words, service time is the average length of time between the start time and end time of a Java method execution. For example, the SQL service time is the time it takes the JDBC method executing the SQL statement to be completed. The service time includes CPU and wait time. The service times are reported in the interval in which they complete execution. Though a contributor's average service time may be very small, the contributor may be called many times. As a result, a contributor's overall contribution to performance may be large even though its average service time is low.
servlet	A class that is loaded only once and for which the application server uses multithreading to process requests. The servlet generates an HTML page that is sent back to the Web browser.
size	In Web, a counter that keeps track of the bytes sent and received.
SLA (Service Level Agreement)	A formal definition of an information system's performance goals. Within Precise, an SLA consists of clauses corresponding to various system activities. Once a system's SLA is defined, its SLA compliance can be analyzed, and breaches can be isolated to identify their causes.
SmartLink	In Insight, a component that provides transactive, correlated information across AppTiers from the user's perspective. Insight SmartLink functions in PeopleSoft and Web applications environments.
SmarTune	In Oracle, SQL Server, and .NET, an algorithm that analyzes performance metrics, identifies and ranks potential problem areas, and provides advice for correcting the problems.
socket	An I/O abstraction layer that enables processes to communicate with each other, regardless of whether they are on the same server. Sockets are bi-directional FIFO pipes that look like regular file I/O to the developer with the abstraction layer handling all of the low-level communication details.
StartPoint	The opening page of Precise. It provides a quick overview of the environment status and links to launch any of the Precise products.
Statistics workspace	In Oracle, DB2, SQL Server, Sybase, J2EE, and Web this workspace presents instance level statistics such as hit ratios, utilization, JMX metrics, logging, and more.
submetric	Also called child metric. In Alerts, the subquery of a parent metric. A child metric gathers its own data and combines it with the data gathered by other child metrics to form the result of the parent metric. Each child metric has its own thresholds and may be enabled or disabled individually.
	See also parent metric.
summary table	A container that stores the information collected by the Precise agents and loads it into the PMDB. Summary tables store the same data at different levels of granularity: time slice, hourly, daily, weekly, and monthly. By storing data in multiple summary tables, Precise can present a detailed view and progressively higher-level views of the same data. Summary tables are particularly useful for data aging. A data purging policy can be implemented for each summary table so that detailed data is retained for short-term historical analyses while more summarized data is used for long-term analyses and trending.

technology	A technology identifies the monitored object. For example, Oracle, SQL Server, Sybase, and DB2 are different technologies, while all Web servers (such as: Apache, IIS, and WebSphere) are defined as a single Web technology. A single monitored object can be monitored by two different technology's Collectors. For example, WebLogic server can be monitored by both a Web Collector and a J2EE Collector.
throughput	The number of average completions per second that are observed during an interval.
time slice	A unit used to break up long sessions into smaller time periods. The length of a time slice is preset and cannot be changed. It represents the maximum time that passes before the data collected can be displayed. For example, if the length of a time slice is 15 minutes, the collection is only updated at 15-minute intervals.
	The length of a time slice is different for each technology: In Oracle, SQL Server, Sybase, DB2, Web, .NET, MQ, Tuxedo, Network, and OS, a time slice is 15 minutes. In J2EE and SmartLink, a time slice is either 30 seconds or 5 minutes. In Precise for SAP, a time slice is 5 minutes.
	See also collector slice size.
URI (Uniform Resource Identifier)	The relative path to a resource after the location (network node) is found.
URL mapping	In Web, a function that defines rules that map URLs (Uniform Resource Locators) with dynamic parameters originating from a specific domain to a format that identifies the Web pages and prevents them from having different URLs.
Web client browser-side	In Web, the gathering of performance data from the browser running on the desktop of the Web application's user, through the static or dynamic instrumentation of Web pages.
Web server	A program that receives client requests for Web pages, retrieves the static pages and/or issues a request for dynamic page creation to an application server, and sends the pages to the client. In Web, a computer that delivers (serves up) Web pages. Every Web server has an IP address and possibly a domain name.
work time	The time spent in the specific entity excluding time spent by called (other) entities. In J2EE, work time of a method is calculated as the service time of that method minus the service time of the methods it invoked.
work type	In Web, a counter that keeps track of the type of the Web instance, such as Web, PeopleSoft, SAP, or Siebel.
workspace	A display unit in Precise products. All workspaces display data from different perspectives. For example, in Oracle, SQL Server, and Sybase, the Current workspace shows information on the sessions currently active in an application, and the Objects workspace displays information on Oracle or SQL Server database objects that can be used to understand relationships and associations between database schema objects.