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.

> 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 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.
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 1 No statement in context error message

See “About the Dashboard tab” on page 46, “About the Current tab” on page 55, “About the Activity tab” on page 74, and “About the Objects tab” on page 105.

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.

Seeking is the recommended way to use indexes.

The following table describes the available operation options.

Table 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.
Nestled 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 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 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 *SQL Server 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 5 Columns table

Column	Description
	Indicates whether the column is part of the selected index and is sorted in ascending order.
	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 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 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 8 Operations

Column	Description
	Locates and highlights the operators in the Execution plan tree that match the selected operation type.
Operation Type	Indicates the 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	Displays the total estimated cost breakdown according to 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.

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 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.

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 Sort order icons

Sort Order Icon	Description
	Indicates whether the data in the column will be stored in the index in ascending order.
	Indicates whether the data in the column will be stored in the index in descending order.
	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.

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.

> 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 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
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 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 13 Estimated Access Plan Changes table

Column	Description
	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 14 Actual Access Plan Changes table

Column	Description
	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 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 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.
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 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 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.

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.

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 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 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.

> 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” on page 164.

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, or
    - 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” on page 167.

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” on page 165.

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.