

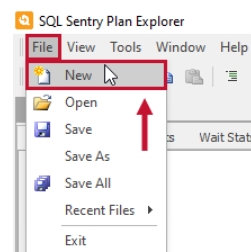
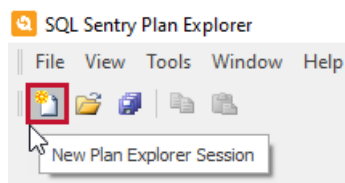
Integrated Plan Explorer Overview

Last Modified on 15 July 2021

Note: The functionality between the standalone and integrated Plan Explorer is nearly the same. See the following information for more details about Plan Explorer.

Note: The default value for the **Query Collection Maximum Text Length** setting is truncated at 10,000 characters. It may be set to a value between 10,000 and 50,000 characters. For more information, see the [Adjusting Captured Text Data](#) topic.

Plan Explorer is designed to make query plan analysis fast and intuitive. There are several ways to start a new **Plan Explorer Session** within SQL Sentry. On the **Top SQL** tab, the **View** button in the **Plan** column opens a **Plan Explorer Session** for the associated completed or running query. Start a new **Plan Explorer Session** with the **New Plan Explorer Session** toolbar button or through the **File** menu.



Plan Explorer Sessions (.pesession) are designed to help you manage a historical record as you fine tune queries. By default, a historical entry is generated as part of the **Plan Explorer Session** during **Estimated** and **Actual Plan** retrieval. For more information, see the [Plan Explorer Sessions](#) topic.

Note: Several of our team members have written about the available features and benefits of using SQL Sentry **Plan Explorer**. Find all of our **Plan Explorer** related posts [on our team blog](#).

General Layout Information

The screen and window layout of **Plan Explorer** is persisted by default, and can be saved to and loaded from a disk. Default horizontal and vertical layout styles are included; the horizontal layout is better suited for shorter queries with wide plans, while the vertical layout is better suited for longer queries. Change the layout by selecting the desired layout button on the toolbar or through the context menu of any section's header.

Entire plans can be saved to and loaded from a file, complete with all statement and plan information. Save a plan by selecting **Save** on the toolbar or through the **File** menu. A **Save As** command is also available from the **File** menu, allowing you to keep different versions of similar plans.



Customizable Tab Layout

The **Plan Explorer** tab layout is completely customizable. Each tab fully undocks from its parent window, allowing you to simultaneously view the **Plan Diagram** alongside any other chosen query detail tabs. To undock a tab from its parent window, select and drag the tab's title bar. Choose to either re-dock the tab with an existing tab, or float the tab in a standalone window. Use the push-pin to unpin the tab from the immediate docked view, and it becomes auto-hidden when not in use.

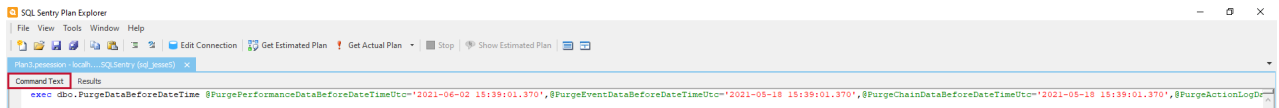
Statements Section

The **Statements Tree** tab shows a tree list representation of the entire query call stack, including all statements, conditional logic structures, looping structures, nested procedure calls, and dynamic SQL calls. Estimated costs (Total, CPU, and IO) and estimated rows are displayed for each statement, and actual costs and rows are shown side-by-side for any statements captured from the **Top SQL** collection or for **Actual Plans** generated within the **Plan Explorer Session**. Any significant differences between estimated and actual rows for captured statements are highlighted, making it easy to spot cases where statistics may be stale.

Statement	Est Cost %	Complete Time	Duration	UDF Duration	CPU	UDF CPU	Est CPU Cost %	Reads	Writes	Est IO Cost %	Est Rows	Actual Rows	Key Lookups	RID Lookups
exec dbo.PurgeDataBeforeDateTime @PurgePerformanceDataBeforeDateTime...	100.0%	0	0	0	0	0	100.0%			100.0%			0	0
SET NOCOUNT ON	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @PurgeBeforeTimestamp = DATEDIFF(second, '20000101 00:00:00', @...	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @PurgeIndexDataBeforeTimestamp = DATEDIFF(second, '20000101 00:...	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @UTCDate = GETUTCDATE()	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @LocalDate = GETDATE()	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @TimeDiff = DATEDIFF(hh, @UTCDate, @LocalDate)	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @PurgePerformanceDataBeforeDateLocal = DATEADD(hh, @TimeDi...	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @OldIndexRequests = DATEADD(HOUR, -1, GETUTCDATE())	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @OldTasksDateTime = DATEADD(DAY, -1, GETUTCDATE()) --Tasks.Task...	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @firstpass = 1	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
WHILE (@@rowcount > 20 OR @firstpass = 1)	0.4%	0	0	0	0	0	0.7%			0.4%			0	0
SET @firstpass = 0	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
WAITFOR DELAY '00:00:00.100'	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
DELETE Tasks.TaskStatus FROM (SELECT TOP 1000 TaskId, OwnerId FR...	0.4%	5	0	0	0	0	0.7%			0.4%	12		0	0
SET @firstpass = 1	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
WHILE (@@rowcount > 20 OR @firstpass = 1)	0.7%	0	0	0	0	0	1.9%			0.7%			0	0
SET @firstpass = 1	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
WHILE (@@rowcount > 20 OR @firstpass = 1)	0.6%	0	0	0	0	0	0.1%			0.6%			0	0
SET @firstpass = 1	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
WHILE (@@rowcount > 20 OR @firstpass = 1)	2.8%	0	0	0	0	0	8.4%			2.8%			0	0
SET @firstpass = 1	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
WHILE (@@rowcount > 20 OR @firstpass = 1)	0.8%	0	0	0	0	0	0.1%			0.8%			0	0
SELECT @MinPerformanceAnalysisSsasTraceDataId = ISNULL(MIN(ID), 0) FR...	0.0%	2	0	0	0	0	0.0%			0.0%	1		0	0

Each of the three sections are automatically synchronized when you select a statement. This includes all the tabs in the **Plan Details** section. Use the **Statements Tree** to quickly find the highest cost statements. Selecting any statement shows the graphical diagram, top operations list, columns list, and tree list view for the statement's execution plan.

The **Command Text** tab allows editing of the SQL for **Estimated** and **Actual Plan** retrieval.



Query Information Section

Command Text Results

Statement	Est Cost %	Complete Time	Duration	UDF Duration	CPU	UDF CPU	Est CPU Cost %	Reads	Writes	Est IO Cost %	Est Rows	Actual Rows	Key Lookups	RID Lookups
SET NOCOUNT ON	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @PurgeBeforeTimestamp = DATEDIFF(second, '20000101 00:00:00', @...	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @PurgeIndexDataBeforeTimestamp = DATEDIFF(second, '20000101 00:...	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @UTCDate = GETUTCDATE()	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @LocalDate = GETDATE()	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @TimeDiff = DATEDIFF(hh, @UTCDate, @LocalDate)	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @PurgePerformanceDataBeforeDateLocal = DATEADD(hh, @TimeDi...	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @OldIndexRequests = DATEADD(HOUR, -1, GETUTCDATE())	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @OldTasksDateTime = DATEADD(DAY, -1, GETUTCDATE()) --Tasks.Task...	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
SET @firstpass = 1	0.0%	0	0	0	0	0	0.0%			0.0%			0	0
WHILE (@@rowcount > 20 OR @firstpass = 1)	0.4%	0	0	0	0	0	0.7%			0.4%			0	0
SET @firstpass = 0	0.0%	0	0	0	0	0	0.0%			0.0%			0	0

Text Data

```

SET @OldIndexRequests = DATEADD(HOUR, -1, GETUTCDATE())
DECLARE @OldTasksDateTime datetime
SET @OldTasksDateTime = DATEADD(DAY, -1, GETUTCDATE())
--Tasks.TaskStatus
SET @firstpass = 1
WHILE (@@rowcount > 20 OR @firstpass = 1)
BEGIN
SET @firstpass = 0
WAITFOR DELAY '00:00:00.100'
DELETE Tasks.TaskStatus
FROM (SELECT TOP 1000 TaskId, OwnerId

```

Plan Diagram

Query Information Tabs

<h3>Query Information tab</h3>	<h3>Image</h3>
<p>The Text Data tab displays the color-coded T-SQL batch or procedure definition of the plan.</p> <p>Save the text with the Save SQL context menu command, or copy the text data to the Command</p>	

Text tab with the **Copy To Command Text** context menu command.

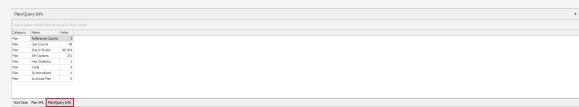
Image

The **Plan XML** tab displays the execution plan in XML format.



Save the plan in **.sqlplan** format, with the **Save Execution Plan** context menu command.

The **Plan/Query Info** tab provides detailed information about the selected query and query plan.



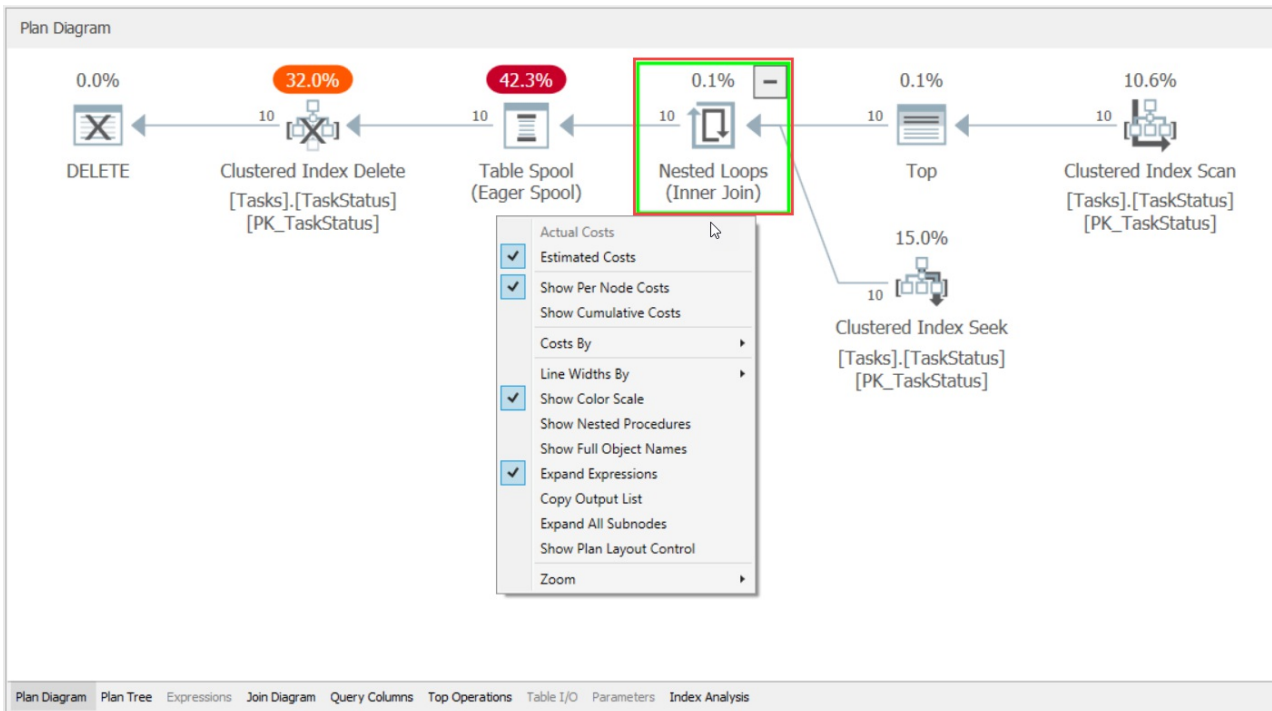
Note: Information displayed in the **Plan/Query Info** tab is only available for queries captured by the SQL Sentry Monitoring Service.

Plan Details Section

When you select a statement in the **Statements Section**, all the tabs in the **Plan Details** section are synchronized with the selected statement. To inspect or make modifications to an index, open the index properties window by right-clicking any index operation node in the **Plan Diagram** tab, or any grid row within the other **Plan Details** tabs; likewise, create new indexes from a separate context item. When you select an operation in any of the **Plan Details** tabs it's automatically synchronized. Select **Copy**, available through the right-click context menu, to copy any of the grid view data in the **Plan Details** section.

The screenshot displays the Plan Details section with three tabs: Command Text, Plan XML, and Plan Diagram. The Command Text tab shows a table of query statements with columns for Est Cost %, Complete Time, Duration, CPU, Est CPU Cost %, Reads, Writes, Est IO Cost %, Est Rows, Actual Rows, Key Lookups, Missing Ind., Clustered I., Index Scan..., Hash Match..., Sort Opera..., and Table Scan... The Plan XML tab shows the XML representation of the execution plan. The Plan Diagram tab shows a graphical representation of the execution plan with nodes and arrows. A context menu is open over the Plan XML tab, showing options: Expand Subtree, Expand All, Collapse All, Edit Connection, Copy, Copy Statement to Command Text, Copy All Statements to Command Text, Call, Row, and All. The 'Copy' and 'Copy Statement to Command Text' options are highlighted.

Plan Diagram Context Menu



The **Plan Diagram** tab has an optimized layout algorithm that renders plans in a much more condensed view than SSMS, so more of the plan fits on the screen without having to zoom out. If needed, zoom in and out by selecting **CTRL + Mouse Wheel**. Optimized plan node labels prevent truncation of object names in most cases. To disable truncation completely so full object names are always visible, select **Show Full Object Names** from the right-click context menu.

The estimated cost of the operation is displayed above each node for maximum readability. These cost labels use color scaling by CPU, IO, or CPU+IO so highest cost operations are instantly obvious, even on larger plans. **CPU + IO** is used by default; change this through the **Costs By** context menu. All costs in the **Plan Diagram** are shown to the first decimal place.

Through the context menu of the **Plan Diagram**, choose to show cumulative costs in lieu of per node costs; when combined with color scaling, this feature makes it easy to see which subtrees are contributing most to the plan cost.

Scale the connector line width by either row or data size with the **Line Widths By** context menu command. The metric you choose to scale by is displayed above all connector lines within the **Plan Diagram**.

Plan Detail Tabs

Plan Details Tabs Grid Legend

Icon	Description
<input checked="" type="checkbox"/>	The check mark indicates a value of Yes.
<input type="checkbox"/>	The empty box indicates a value of No.



The filled box indicates a value of Not Applicable.
Description

The **Plan Tree** tab is a tree representation of the plan that shows all operations and associated metrics. Use the arrows to expand and collapse sections of the **Plan Tree**. Right-click any column header, and use the **Column Chooser** command to access additional plan metrics. Any significant differences between estimates and actuals are highlighted, as are possible problematic operations like scans and bookmark lookups.

Operation	Object	Est Cost %	Est Subtree Cost %	Est Rows	Est Executions	Est CPU Cost	Est IO Cost	Est Data Size	Parallel	Ordered	Est Rebinds	Est Rewinds	Partitioned
Index Scan	[SQLSentry].[dbo].[EventSourceHistory].[IX_P...	0.1%	0.0%	2,053	1	0.0024153	0.0083102	30,795	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	0	<input type="checkbox"/>
SELECT		0.0%	0.0%	1	0	0.0000000	0.0000000	0	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Compute Scalar		0.0%	0.0%	1	1	0.0000001	0.0000000	15	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Stream Aggregate		0.0%	0.0%	1	1	0.0000011	0.0000000	15	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Top		0.0%	0.0%	1	1	0.0000001	0.0000000	15	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Scan	[SQLSentry].[dbo].[EventSourceHistory].[PK_E...	0.0%	0.0%	1	1	0.0024153	0.008678	15	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	0	<input type="checkbox"/>
DELETE		0.0%	0.0%	6	0	0.0000000	0.0000000	0	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Delete	[SQLSentry].[dbo].[EventSourceHistoryDetail]....	0.0%	0.0%	6	1	0.0000405	0.0700000	52	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Table Spool (Eager Spool)		0.1%	0.0%	6	1	0.0001022	0.0131249	87	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Nested Loops (Inner Join)		0.0%	0.0%	6	1	0.0000242	0.0000000	87	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Top		0.0%	0.0%	6	1	0.0000005	0.0000071	87	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Index Seek	[SQLSentry].[dbo].[EventSourceHistoryDetail]....	0.0%	0.0%	6	1	0.0001659	0.0031250	133	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Seek	[SQLSentry].[dbo].[EventSourceHistoryDetail]....	0.1%	0.0%	6	6	0.0009145	0.0060232	87	<input type="checkbox"/>	<input checked="" type="checkbox"/>	5	0	<input type="checkbox"/>
INSERT		0.0%	0.0%	1	0	0.0000000	0.0000000	0	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Table Insert	[tempdb].[dbo].[#tempMinRemoteSequenceIDs]	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Stream Aggregate		0.0%	0.0%	1	1	0.0000001	0.0000000	23	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Index Seek	[SQLSentry].[dbo].[EventSourceHistoryDetail]....	0.0%	0.0%	5	1	0.0001627	0.0031250	120	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	0	<input type="checkbox"/>
UPDATE		0.0%	0.0%	1	0	0.0000000	0.0000000	0	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Table Update	[tempdb].[dbo].[#tempMinRemoteSequenceIDs]	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Compute Scalar		0.0%	0.0%	1	1	0.0000001	0.0000000	23	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Nested Loops (Left Outer Join)		0.0%	0.0%	1	1	0.0000042	0.0000000	23	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Table Scan	[tempdb].[dbo].[#tempMinRemoteSequenceIDs]	0.0%	0.0%	1	1	0.0001581	0.0031250	31	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	0	<input type="checkbox"/>
Top		0.0%	0.0%	1	1	0.0000001	0.0000000	15	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>

The **Top Operations** tab contains a sortable list view of all plan operations that's sorted by total cost descending by default so you can immediately see which operations are the highest cost.

Operation	Object	Est Cost %	Est Subtree Cost %	Est Rows	Est Executions	Est CPU Cost	Est IO Cost	Est Data Size	Parallel	Ordered	Est Rebinds	Est Rewinds	Partitioned
Index Scan	[SQLSentry].[dbo].[EventSourceHistory].[IX_P...	0.1%	0.0%	2,053	1	0.0024153	0.0083102	30,795	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Insert	[tempdb].[dbo].[#tempProcessHistoryID].[PK_...	0.1%	0.0%	523	1	0.0005226	0.0100000	4,703	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Seek	[SQLSentry].[dbo].[EventSourceHistoryDetail]....	0.1%	0.0%	27	27	0.0042687	0.0062500	405	<input type="checkbox"/>	<input checked="" type="checkbox"/>	25	0	<input type="checkbox"/>
Index Scan	[SQLSentry].[dbo].[PerformanceAnalysisDevice...	0.1%	0.0%	523	1	0.0020732	0.0083102	9,929	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Merge Join (Left Semi Join)		0.1%	0.0%	242	1	0.0103137	0.0000025	3,075	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Index Scan	[SQLSentry].[dbo].[PerformanceAnalysisSqlInd...	0.1%	0.0%	1,613	1	0.0001933	0.0083102	20,969	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Delete	[SQLSentry].[tasks].[TaskStatus].[PK_TaskSta...	0.1%	0.0%	12	1	0.0000120	0.0100000	108	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Insert	[SQLSentry].[dbo].[PerformanceAnalysisSqlFile...	0.1%	0.0%	9	1	0.0000090	0.0100000	81	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Insert	[SQLSentry].[dbo].[PerformanceAnalysisDevice...	0.1%	0.0%	7	1	0.0000070	0.0100000	63	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Insert	[SQLSentry].[dbo].[PerformanceAnalysisTrace...	0.1%	0.0%	7	1	0.0000066	0.0100000	59	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Insert	[SQLSentry].[dbo].[PerformanceAnalysisDevice...	0.1%	0.0%	3	1	0.0000030	0.0100000	27	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Delete	[SQLSentry].[dbo].[PerformanceAnalysisFragm...	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Table Update	[tempdb].[dbo].[#tempMinRemoteSequenceIDs]	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Index Delete	[SQLSentry].[dbo].[PerformanceAnalysisSeasP...	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Index Delete	[SQLSentry].[dbo].[PerformanceAnalysisSeasM...	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Index Delete	[SQLSentry].[dbo].[PerformanceAnalysisSeasC...	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Index Delete	[SQLSentry].[dbo].[PerformanceAnalysisSeasC...	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Index Delete	[SQLSentry].[dbo].[PerformanceAnalysisSeasC...	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Delete	[SQLSentry].[VM].[AuditVirtualMachineHostSyst...	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Delete	[SQLSentry].[SqPerf].[History].[PK_Synchistor...	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>
Clustered Index Delete	[SQLSentry].[SqPerf].[SessionTableStatus].[P...	0.1%	0.0%	1	1	0.0000010	0.0100000	9	<input type="checkbox"/>	<input type="checkbox"/>	0	0	<input type="checkbox"/>

The **Query Columns** tab shows exactly how data is accessed for each column used by the query, including the associated operation and estimated rows. This view makes it easy to identify bookmark lookups or index scans occurring because indexes are non-covering.

Database	Schema	Table	Column	Operation	Index	Est Rows	Est Data Size	Filters	Group By
		#tempMirRemoteSequenceIDs	EventSourceID	Table Scan		1	31		<input type="checkbox"/>
		#tempMirRemoteSequenceIDs	MirRemoteSequenceID	Table Scan		1	31		<input type="checkbox"/>
		#tempProcessHistoryID	ID	Clustered Index Scan	[PK_...#tempPro_3214EC2782D5B05A]	1	15		<input type="checkbox"/>
		[#HistoryMetaTables]	name	Table Scan		1	4,035		<input type="checkbox"/>
[SQL Sentry]	[dbo]	[AlertingChannelLog]	ID	Hash Match (Inner Join)		70	1,049	[SQL Sentry].[dbo].[AlertingChannelLog].[ID] = ...	<input type="checkbox"/>
[SQL Sentry]	[dbo]	[AlertingChannelLog]	ID	Index Seek	[IX_Covering]	50	756		<input type="checkbox"/>
[SQL Sentry]	[dbo]	[AlertingChannelLog]	NormalizedStartTimeUtc	Index Seek	[IX_Covering]	50	756	[SQL Sentry].[dbo].[AlertingChannelLog].[Norm...	<input type="checkbox"/>
[SQL Sentry]	[dbo]	[AlertingChannelLog]	ID	Index Scan	[IX_Covering]	669	10,035		<input type="checkbox"/>
[SQL Sentry]	[dbo]	[AlertingChannelLog]	ID	Hash Match (Inner Join)		1	11	[SQL Sentry].[dbo].[AlertingChannelLog].[ID] = ...	<input type="checkbox"/>
[SQL Sentry]	[dbo]	[AlertingChannelLogNote]	ID	Clustered Index Scan	[PK_AlertingChannelLogNote]	1	19		<input type="checkbox"/>
[SQL Sentry]	[dbo]	[AlertingChannelLogNote]	AlertingChannelLogID	Clustered Index Scan	[PK_AlertingChannelLogNote]	1	19		<input type="checkbox"/>
[SQL Sentry]	[VM]	[AuditVirtualMachineHostSystemChange]	ID	Clustered Index Scan	[PK_AuditVirtualMachineHostSystemChange]	1	19		<input type="checkbox"/>
[SQL Sentry]	[VM]	[AuditVirtualMachineHostSystemChange]	Timestamp	Clustered Index Scan	[PK_AuditVirtualMachineHostSystemChange]	1	19	< @PurgePerformanceDataBeforeDateTimeLocal	<input type="checkbox"/>
[SQL Sentry]	[VM]	[AuditVirtualMachineHostSystemChange]	ID	Clustered Index Seek	[PK_AuditVirtualMachineHostSystemChange]	1	11		<input type="checkbox"/>
[SQL Sentry]	[dbo]	[BlockChain]	ID	Index Seek	[IX_BlockData_LogTimeUtc]	1	15		<input type="checkbox"/>
[SQL Sentry]	[dbo]	[BlockChain]	NormalizedStartTime	Index Seek	[IX_BlockData_LogTimeUtc]	1	15	[SQL Sentry].[dbo].[BlockChain].[NormalizedSta...	<input type="checkbox"/>
[SQL Sentry]	[dbo]	[BlockChain]	ID	Clustered Index Seek	[PK_BlockChain]	1	15		<input type="checkbox"/>
[SQL Sentry]	[dbo]	[BlockChain]	ID	Index Scan	[IX_BlockChainMessageStatus]	1	15		<input type="checkbox"/>
[SQL Sentry]	[dbo]	[BlockChainDetail]	ID	Index Seek	[IX_BlockedBySpid]	1	15		<input type="checkbox"/>
[SQL Sentry]	[dbo]	[BlockChainDetail]	BlockChainID	Index Seek	[IX_BlockedBySpid]	1	15	[SQL Sentry].[dbo].[BlockChainDetail].[BlockCha...	<input type="checkbox"/>

The **Parameter** tab shows all statement parameters with runtime and compiled values.

Parameter	Compiled Value	Runtime Value
[@Counter]		@Counter - 1

When an **Actual Plan** is captured or generated, the **Table I/O** tab is shown. This tab breaks down all the reads from a query by object, making it easy to see where most of your I/O is coming from.

Table	LOB Read-Ahead Reads	LOB Physical Reads	LOB Logical Reads	Read-Ahead Reads	Physical Reads	Logical Reads	Scan Count	Rowgroups Read	Rowgroups Skipped
columns	0	0	20	0	0	0	1	1	0
tables	0	0	0	0	0	21	1		
Totals	0	0	20	0	0	21	2	1	0

The **Wait Stats** tab was introduced as part of the feature set of SQL Sentry 7.2. When an **Actual Plan** is retrieved with related wait stats, the **Wait Stats** tab displays next to the **Results** tab. The wait type is shown along with any associated wait time or signal time. The **Wait Stats** tab helps you in identifying potential resource bottlenecks, including those related to memory pressure, CPU pressure, disk I/O, and networking.

Wait Type	Wait Time	Signal Time
WAITFOR	7,657	8
WRITELOG	1	0

Note: **Wait Stats** are available for queries executed against servers running SQL Server 2008 or newer.

Plan History

SQL Sentry tracks all plan versions for a query over time, so you can easily determine when a plan has changed and caused query performance problems. The plans are the **Estimated Plans** collected at the time of query execution. For more information, see the [Query Plans](#) topic.