Execution plan graph operators table

Operators describe how SQL Server executes a query or a Data Manipulation Language (DML) statement. The query optimizer uses operators to build a query plan to create the result specified in the query, or to perform the operation specified in the DML statement. The query plan is a tree consisting of physical operators.

This is the table of descriptions to be used in the properties dialog showing the operator details in the execution plan graphical view.

Operator	Graphical execution plan icon	Text
Aggregate	None	The **Aggregate** operator is a physical or logical operator that computes a new value using the SQL Server functions MIN, MAX, SUM, COUNT or AVG.
Arithmetic Expression		The **Arithmetic Expression** operator computes a new value from existing values in a row. **Arithmetic Expression** is not used in SQL Server 2014.
Assert		The **Assert** operator verifies a condition. The **Assert** operator is a physical operator.
Assign		The **Assign** operator assigns a value to a variable
Asnyc Concat	None	The **Asnyc Concat** operator is used in remote distributed queries to get output rows from remote child nodes to send to the parent node.
Bitmap		The **Bitmap** operator is used to apply filtering to parallel query plans. **Bitmap** is a physical operator.
Bitmap Create		The **Bitmap Create** operator is a logical operator that shows where bitmaps are built.
Bookmark Lookup		The **Bookmark Lookup** operator uses a bookmark (row ID or clustering key) to look up the corresponding row in the table or clustered index. **Bookmark Lookup** is not used in SQL Server 2014. The **Key Lookup** operator also provides this functionality.
Branch Repartition	None	The **Branch Repartition** operator is a logical operator showing where iterators could be executed by parallel threads.
Broadcast	None	The **Broadcast** operator sends the set of input rows to multiple consumers.
Build Hash		The **Build Hash** operator indicates that a batch has table has been built.
Cache	None	The **Cache** operator is a logical operator that caches a single row of data. **Cache** is not used in SQL Server 2014.
Clustered Index Delete		The **Clustered Index Delete** operator deletes rows from a clustered index. **Clustered Index Delete** is a physical operator.
Clustered Index Insert		The **Clustered Index Insert** operator inserts rows into a clustered index. **Clustered Index Insert** is a physical operator.
Clustered Index Merge		The **Clustered Index Merge** operator merges a data stream and a clustered index. **Clustered Index Merge** is a physical operator.
Clustered Index Scan		The **Clustered Index Scan** operator scans a clustered index. It returns only rows matching the WHERE clause and sorts the results based on the ORDERED clause.
Clustered Index Seek		The **Clustered Index Seek** operator retrieves rows from a clustered index using the seeking ability of indexes.

Clustered Index Update		The **Clustered Index Update** operator updates rows in the clustered index. It only updates rows matching the WHERE clause.
Collapse		The **Collapse** operator merges separate operations into a single more efficient operation.
Columnstore Index Scan		The **Columnstore Index Scan** operator scans the columnstore index specified in the query execution plan argument.
Compute Scalar		The **Compute Scalar** operator returns a computer scalar value from an evaluated expression.
Concatenation		The **Concatenation** operator returns rows scanned from multiple inputs. This is used for statements like UNION ALL.
Constant Scan		The **Constant Scan** operator adds rows into a query.
Convert		The **Convert** operator converts one scalar type to another.
Cross Join	None	The **Cross Join** operator is a logical operator that joins each row from input with each row from another input.
catchall		The **catchall** operator is a placeholder icon when an operator does not match any other query operators.
Cursor	None	The **Cursor** operator describes the execution of a query or update that use cursor operations.
Declare		The **Declare** operator allocates a local variable.
Delete		The **Delete** operator deletes rows from an object.
Deleted Scan		The **Deleted Scan** operator scans the deleted table within a trigger.
Distinct	None	The **Distinct** operator removes duplicate rows.
Distinct Sort	None	The **Distinct Sort** operator removes duplicate rows and sorts the resulting set of rows.
Distribute Streams		The **Distribute Streams** operator is used to break records from a single input to multiple output streams in a parallel query plan.
Dynamic		The **Dynamic** operator uses a cursor to see all changes made by others.
Eager Spool		The **Eager Spool** operator stores each row in the input to tempdb so that this cached data can be used if the operator is rewound.
Fetch Query		The **Fetch Query** operator returns rows from a fetch issued against a cursor.
Filter		The **Filter** operator returns only rows that match a filter expression.
Flow Distinct	None	The **Flow Distinct** operator removes duplicate rows but returns each row as it is processed.
Full Outer Join	None	The **Full Outer Join** operator is a logical operator that implements an OUTER JOIN. Results in all matching rows from two streams plus rows for each row in the streams that did not have matches.

Gather Streams		The **Gather Streams** operator is used to consume multiple input streams and combine them into a single output stream in parallel query plans.
Hash Match		The **Has Match** operator creates a hash table from the rows in the build input.
Inner Join	None	The **Inned Join** operator returns rows that satisfy the JOIN of the first (top) input with the second (bottom) input.
Insert		The **Insert** logical operator inserts input rows into the object specified in the Argument column.
Inserted Scan		The **Inserted Scan** operator scans the inserted table.
Intrinsic		The **Intrinsic** operator runs an internal Transact-SQL function.
Iterator		The **Iterator** operator is a placeholder icon used when no match is found for the iterator operation.
Key Lookup		The KEY LOOKUP operator signals a lookup using a bookmark on a table with a clustered index.
Keyset		The KEYSET operator uses a cursor that can only see updates and not inserts.
Language Element		The LANGUAGE ELEMENT operator is a placeholder icon when no matching language constructs can be found.
Lazy Spool		The LAZY SPOOL operator stores each row in the input to tempdb so that this cached data can be used if the operator is rewound. The rows are only copied to tempdb as they are processed.
Left Anti Semi Join	None	The LEFT ANTI SEMI JOIN operator returns rows from the first input where there is no match in the second input.
Left Outer Join	None	The LEFT OUTER JOIN operator returns rows from the first input that satisfy the join with the second input.
Left Semi Join	None	The LEFT SEMI JOIN operator returns rows from the first input that have a match with the second input.
Log Row Scan		The LOG ROW SCAN operator scans the transaction log.
Merge Interval		The MERGE INTERVAL operator merges multiple intervals to create a minimal set of non-overlapping intervals.
Merge Join		The MERGE JOIN operator performs one of the many possible join operations.
Nested Loops		The NESTED LOOPS operator perform the logical operations to satisfy many join operations that need a loop to search the inner table for rows in the outer table.
Nonclustered Index Delete		The NONCLUSTERED INDEX DELETE operator deleted rows from a non-clustered index.
Index Insert		The INDEX INSERT operator inserts rows into a non-clustered index.
Index Scan		The INDEX SCAN operator scans a non-clustered index. It returns only rows matching the WHERE clause
Index Seek		The INDEX SEEK operator retrieves rows from a non-clustered index using the seeking ability of indexes.

Index Spool		The INDEX SPOOL operator copies input rows to tempdb and builds a non-clustered index for these rows.
Noncluster ed Index Update		The NONCLUSTERED INDEX UPDATE operator updates rows from its input in the non-clustered index.
Online Index Insert		The ONLINE INDEX INSERT operator is a physical operator indicating that an index create, alter or drop is performed online.
Parallelism	None	The PARALLELISM operator the logical operations of distribute, gather or repartition streams.
Parameter Table Scan		The PARAMETER TABLE SCAN operator scans a table that acts as a parameter of the query.
Partial Aggregate	None	The PARTIAL AGGREGATE operator is a logical operator that aggregates input rows to prevent writing to disk in parallel plans.
Population Query		The POPULATION QUERY operator populates a cursor's work table.
Refresh Query		The REFRESH QUERY operator fetches current data for rows.
Remote Delete		The REMOTE DELETE operator deletes rows from an remote object.
Remote Index Scan		The REMOTE INDEX SCAN operator scans a remote index.
Remote Index Seek		The REMOTE INDEX SEEK operator retrieves rows using a remote index.
Remote Insert		The REMOTE INSERT operator inserts rows into a remote object.
Remote Query		The REMOTE QUERY operator submits a query to a remote source.
Remote Scan		The REMOTE SCAN operator scans a remote object.
Remote Update		The REMOTE UPDATE operator updates a remote object.
Repartition Streams		The REPARTITION STREAMS operator creates multiple out streams from multiple input streams while applying a bitmap filter.
Result		The RESULT operator is the query plan return data.
RID Lookup		The RID LOOKUP operator is a bookmark lookup on a heap.
Right Anti Semi Join	None	The RIGHT ANTI SEMI JOIN operator returns rows from the second input where the is no match in the first input.
Right Outer Join	None	The RIGHT OUTER JOIN operator returns rows from the second input that satisfy the join with the first input.
Right Semi Join	None	The RIGHT SEMI JOIN operator returns rows from the second input that have a match with the first input.

Row Count Spool		The ROW COUNT SPOOL operator returns empty rows for each row in the input stream.
Segment		The SEGMENT operator uses the value of columns to divide the input set into segments.
Segment Repartition	None	The SEGMENT REPARTITION operator marks the boundaries of regions whose iterators can be run in parallel threads.
Sequence		The SEQUENCE operator executes each input in sequence.
Sequence Project		The SEQUENCE PROJECT operator adds columns to the input set, divides the input set into segments and outputs one segment at a time.
Snapshot		The SNAPSHOT operator creates a cursor that cant see changes by others.
Sort		The SORT operator sorts incoming rows.
Split		The SPLIT operator creates a delete and insert operation out of each update operation.
Spool		The SPOOL operator saves an intermediate query to tempdb.
Stream Aggregate		The STREAM AGGREGATE operator groups rows by columns and calculates aggregate expressions.
Switch		The SWITCH operator copies the appropriate input stream to the output stream by evaluating an expression.
Table Delete		The TABLE DELETE operator deletes rows from a table.
Table Insert		The TABLE INSERT operator inserts rows into a table.
Table Merge		The TABLE MERGE operator applies a merge data stream to a heap.
Table Scan		The TABLE SCAN operator retrieves rows from a table.
Table Spool		The TABLE SPOOL operator scans the input and places the rows into tempdb.
Table Update		The TABLE UPDATE operator updates rows in a table.
Table-valued Function		The TABLE-VALUED FUNCTION operator evaluates a table-valued function and stores the resulting rows in tempdb.
Top		The TOP operator returns only the specified number of rows from the input.
Top N Sort	None	The TOP N SORT operator returns only the specified number of rows from the input and sorts them.
UDX		The UDX operator implements XQuery and XPath operations.
Union	None	The UNION operator combines multiple inputs and removes duplicates.

Update		The UPDATE operator updates a specified object from the rows in its input.
While		The WHILE operator represents a SQL while loop.
Window Spool		The WINDOW SPOOL operator expands the input rows into sets of rows that represent the window associated with the row.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal