Collapse All Code: All

DAX Operator Reference for PowerPivot

See Also Send Feedback

The Data Analysis Expression (DAX) language uses operators to create expressions that compare values, perform arithmetic calculations, or work with strings. This section describes the use of each operator.

Types of Operators

There are four different types of calculation operators: arithmetic, comparison, text concatenation, and logical.

Arithmetic Operators

To perform basic mathematical operations such as addition, subtraction, or multiplication; combine numbers; and produce numeric results, use the following arithmetic operators.

Arithmetic operator	Meaning	Example
+ (plus sign)	Addition	3+3
– (minus sign)	Subtraction/Sign	3–1–1
* (asterisk)	Multiplication	3*3
/ (forward slash)	Division	3/3
^ (caret)	Exponentiation	16^4

Note

The plus sign can function both as a *binary operator* and as a *unary operator*. A binary operator requires numbers on both sides of the operator and performs addition. When you use values in a DAX formula on both sides of the binary operator, DAX tries to cast the values to numeric data types if they are not already numbers. In contrast, the unary operator can be applied to any type of argument. The plus symbol does not affect the type or value and is simply ignored, whereas the minus operator creates a negative value, if applied to a numeric value.

Comparison Operators

You can compare two values with the following operators. When two values are compared by using these operators, the result is a logical value, either TRUE or FALSE.

Comparison operator	Meaning	Example
=	Equal to	[Region] = "USA"
>	Greater than	[Sales Date] > "Jan 2009"
<	Less than	[Sales Date] < "Jan 1 2009"
>=	Greater than or equal to	[Amount] >= 20000
<=	Less than or equal to	[Amount] <= 100
<>	Not equal to	[Region] <> "USA"

Text Concatenation Operator

Use the ampersand (&) to join, or concatenate, two or more text strings to produce a single piece of text.

Text operator	Meaning	Example
& (ampersand)	Connects, or concatenates, two values to produce one continuous text value	[Region] & ", " & [City]

Logical Operators

Use logical operators (&&) and (||) to combine expressions to produce a single result.

Text operator	Meaning	Examples
&& (double ampersand)	Creates an AND condition between two expressions that each have a Boolean result. If both expressions return TRUE, the combination of the expressions also returns TRUE; otherwise the combination returns FALSE.	([Region] = "France") && ([BikeBuyer] = "yes"))
|| (double pipe symbol)	Creates an OR condition between two logical expressions. If either expression returns TRUE, the result is TRUE; only when both expressions are FALSE is the result FALSE.	(([Region] = "France") || ([BikeBuyer] = "yes"))

Operators and Precedence Order

In some cases, the order in which calculation is performed can affect the return value; therefore, it is important to understand how the order is determined and how you can change the order to obtain the desired results.

Calculation Order

An expression evaluates the operators and values in a specific order. All expressions always begin with an equal sign (=). The equal sign indicates that the succeeding characters constitute an expression.

Following the equal sign are the elements to be calculated (the operands), which are separated by calculation operators. Expressions are always read from left to right, but the order in which the elements are grouped can be controlled to some degree by using parentheses.

Operator Precedence

If you combine several operators in a single formula, the operations are ordered according to the following table. If the operators have equal precedence value, they are ordered from left to right. For example, if an expression contains both a multiplication and division operator, they are evaluated in the order that they appear in the expression, from left to right.

Operator	Description
^	Exponentiation
–	Sign (as in –1)
* and /	Multiplication and division
+ and –	Addition and subtraction
&	Connects two strings of text (concatenation)
=< ><=>=<>	Comparison

Using Parentheses to Control Calculation Order

To change the order of evaluation, you should enclose in parentheses that part of the formula that must be calculated first. For example, the following formula produces 11 because multiplication is calculated before addition. The formula multiplies 2 by 3, and then adds 5 to the result.

Copy Code

```
=5+2*3
```

In contrast, if you use parentheses to change the syntax, the order is changed so that 5 and 2 are added together, and the result multiplied by 3 to produce 21.

Copy Code

```
=(5+2)*3
```

In the following example, the parentheses around the first part of the formula force the calculation to evaluate the expression `(3 + 0.25)` first and then divide the result by the result of the expression, `(3 – 0.25)`.

Copy Code

```
=(3 + 0.25)/(3 – 0.25)
```

In the following example, the exponentiation operator is applied first, according to the rules of precedence for

operators, and then the sign operator is applied. The result for this expression is -4.

Copy Code

```
=-2^2
```

To ensure that the sign operator is applied to the numeric value first, you can use parentheses to control operators, as shown in the following example. The result for this expression is 4.

Copy Code

```
= (-2)^2
```

Compatibility Notes

DAX easily handles and compares various data types, much like Microsoft Excel. However, the underlying computation engine is based on SQL Server Analysis Services and provides additional advanced features of a relational data store, including richer support for date and time types. Therefore, in some cases the results of calculations or the behavior of functions may not be the same as in Excel. Moreover, DAX supports more data types than does Excel. This section describes the key differences.

Coercing Data Types of Operands

In general, the two operands on the left and right sides of any operator should be the same data type. However, if the data types are different, DAX will convert them to a common data type for comparison, as follows:

1. First, both operands are converted to the largest possible common data type.
2. Next, the operands are compared.

For example, suppose you have two numbers that you want to combine. One number results from a formula, such as `=[Price] * .20`, and the result may contain many decimal places. The other number is an integer that has been provided as a string value.

In this case, DAX will convert both numbers to real numbers in a numeric format, using the largest numeric format that can store both kinds of numbers. Then DAX will compare the values.

In contrast, Excel tries to compare values of different types without first coercing them to a common type. For this reason, you may see different results in DAX than in Excel for the same comparison expression.

Data Types used in DAX	Data Types used in Excel
Numbers (I8, R8)	Numbers (R8)
Boolean	Boolean
String	String
DateTime	Variant
Currency	Currency

For more information about implicit data type conversion, see Data Types Supported in PowerPivot Workbooks.

Differences in Precedence Order

The precedence order of operations in DAX formulas is basically the same as that used by Microsoft Excel, but some Excel operators are not supported, such as percent. Also, ranges are not supported.

Therefore, whenever you copy and paste formulas from Excel, be sure to review the formula carefully, as some operators or elements in the formulas may not be valid. When there is any doubt about the order in which operations are performed, we recommend that you use parentheses to control the order of operations and remove any ambiguity about the result.

See Also

Concepts

DAX Syntax Specification for PowerPivot

Data Analysis Expressions (DAX) Overview