# OEE Studio v3.0

## Expression Operators, Functions and Constants

This page lists operators and functions supported by expressions. It also provides information on how constants can be specified in expressions.

An expression is a string that, when parsed and processed, evaluates some value. Expressions consist of column/field names, constants, operators and functions.

## Operators

| Operator | Description | Example |
|---|---|---|
| + | Adds the value of one numeric expression to another or concatenates two strings. | [FirstName] + ' ' + [LastName]<br>[UnitPrice] + 4 |
| - | Finds the difference between two numbers. | [Price1] - [Price2] |
| * | Multiplies the value of two expressions. | [Quantity] * [UnitPrice] * (1 - [BonusAmount]) |
| / | Divides the first operand by the second. | [Quantity] / 2 |
| % | Returns the remainder (modulus) obtained by dividing one numeric expression into another. | [Quantity] % 3 |
| \| | Compares each bit of its first operand to the corresponding bit of its second operand. If either bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0. | [Flag1] \| [Flag2] |
| & | Performs a bitwise logical AND operation between two integer values. | [Flag] & 10 |
| ^ | Performs a logical exclusion on two Boolean expressions or a bitwise exclusion on two numeric expressions. | [Flag1] ^ [Flag2] |
| == | Returns true if both operands have the same value; otherwise, it returns false. | [Quantity] == 10 |
| != | Returns true if the operands do not have the same value; otherwise, it returns false. | [Country] != 'France' |


Deb-Tech Systems, Inc. ~ Algonquin, Illinois ~ 847.854.3148 (o) Debbie.Olk@debtechsystems.com
www.OEEStudio.com

| | | |
|---|---|---|
| < | Less than operator. Used to compare expressions. | [UnitPrice] < 20 |
| <= | Less than or equal to operator. Used to compare expressions. | [UnitPrice] <= 20 |
| >= | Greater than or equal to operator. Used to compare expressions. | [UnitPrice] > 30 |
| > | Greater than operator. Used to compare expressions. | [UnitPrice] >= 30 |
| In (,,,) | Tests for the existence of a property in an object. | [Country] In ('USA', 'UK', 'Italy') |
| Like | Compares a string against a pattern. If the value of the string matches the pattern, result is true. If the string does not match the pattern, result is false. If both string and pattern are empty strings, the result is true. | [Name] Like 'An%' |
| Between (,) | Specifies a range to test. Returns true if a value is greater than or equal to the first operand and less than or equal to the second operand. | [Quantity] Between (10, 20) |
| And | Performs a logical conjunction on two expressions. | [InStock] And ([ExtendedPrice]> 100) |
| Or | Performs a logical disjunction on two Boolean expressions. | [Country]=='USA' Or [Country]=='UK' |
| Not | Performs logical negation on an expression. | Not [InStock] |

# Functions

## Date-time Functions

| Function | Description | Example |
|---|---|---|
| AddDays(DateTime, DaysCount) | Returns a date-time value that is the specified number of days away from the specified DateTime. | AddDays([OrderDate], 30) |
| AddHours(DateTime, HoursCount) | Returns a date-time value that is the specified number of hours away from the specified DateTime. | AddHours([StartTime], 2) |
| AddMilliSeconds(DateTime, MilliSecondsCount) | Returns a date-time value that is the specified number of milliseconds away from the specified DateTime. | AddMilliSeconds(([StartTime], 5000)) |

| | | |
|---|---|---|
| AddMinutes(DateTime, MinutesCount) | Returns a date-time value that is the specified number of minutes away from the specified DateTime. | AddMinutes([StartTime], 30) |
| AddMonths(DateTime, MonthsCount) | Returns a date-time value that is the specified number of months away from the specified DateTime. | AddMonths([OrderDate], 1) |
| AddSeconds(DateTime, SecondsCount) | Returns a date-time value that is the specified number of seconds away from the specified DateTime. | AddSeconds([StartTime], 60) |
| AddTicks(DateTime, TicksCount) | Returns a date-time value that is the specified number of ticks away from the specified DateTime. | AddTicks([StartTime], 5000) |
| AddTimeSpan(DateTime, TimeSpan) | Returns a date-time value that is away from the specified DateTime for the given TimeSpan. | AddTimeSpan([StartTime], [Duration]) |
| AddYears(DateTime, YearsCount) | Returns a date-time value that is the specified number of years away from the specified DateTime. | AddYears([EndDate], -1) |
| GetDate(DateTime) | Extracts a date from the defined DateTime. | GetDate([OrderDateTime]) |
| GetDay(DateTime) | Extracts a day from the defined DateTime. | GetDay([OrderDate]) |
| GetDayOfWeek(DateTime) | Extracts a day of the week from the defined DateTime. | GetDayOfWeek([OrderDate]) |
| GetDayOfYear(DateTime) | Extracts a day of the year from the defined DateTime. | GetDayOfYear([OrderDate]) |
| GetHour(DateTime) | Extracts an hour from the defined DateTime. | GetHour([StartTime]) |
| GetMilliSecond(DateTime) | Extracts milliseconds from the defined DateTime. | GetMilliSecond([StartTime]) |
| GetMinute(DateTime) | Extracts minutes from the defined DateTime. | GetMinute([StartTime]) |
| GetMonth(DateTime) | Extracts a month from the defined DateTime. | GetMonth([StartTime]) |
| GetQuarter() | Extracts the quarter from the defined DateTime | GetQuarter([StartTime]) |

| | | |
|---|---|---|
| GetSecond(DateTime) | Extracts seconds from the defined DateTime. | GetSecond([StartTime]) |
| GetTimeOfDay(DateTime) | Extracts the time of the day from the defined DateTime, in ticks. | GetTimeOfDay([StartTime]) |
| GetWeekOfYear() | Extracts the week of the year from the defined DateTime. | GetWeekOfYear([EndTime]) |
| GetYear(DateTime) | Extracts a year from the defined DateTime. | GetYear([StartTime]) |
| Now() | Returns the current system date and time. | AddDays(Now(), 5) |
| Today() | Returns the current date. Regardless of the actual time, this function returns midnight of the current date. | AddMonths(Today(), 1) |
| UtcNow() | Returns the current system date and time that is expressed as Coordinated Universal Time (UTC). | AddDays(UtcNow(), 7) |

## Logical Functions

| Function | Description | Example |
|---|---|---|
| Iif(Expression, TruePart, FalsePart) | Returns either TruePart or FalsePart, depending on the evaluation of the Boolean Expression. | Iif([Quantity>=10], 10, 0 ) |
| IsNull(Value) | Returns True if the specified Value is NULL. | IsNull([OrderDate]) |
| IsNull(Value1, Value2) | Returns Value1 if it is not set to NULL; otherwise, Value2 is returned. | IsNull([ShipDate], [RequiredDate]) |
| IsNullOrEmpty(String) | Returns True if the specified String object is NULL or an empty string; otherwise, False is returned. | IsNullOrEmpty([ProductName]) |

## Math Functions

| Function | Description | Example |
|---|---|---|
| Abs(Value) | Returns the absolute positive value of the given numeric expression. | Abs(1 - [Discount]) |
| Acos(Value) | Returns the arccosine of a number (the angle, in radians, whose cosine is the given float expression). | Acos([Value]) |

| | | |
|---|---|---|
| Asin(Value) | Returns the arcsine of a number (the angle, in radians, whose sine is the given float expression). | Asin([Value]) |
| Atn(Value) | Returns the arctangent of a number (the angle, in radians, whose tangent is the given float expression). | Atn([Value]) |
| Atn2(Value1, Value2) | Returns the angle, whose tangent is the quotient of two specified numbers, in radians. | Atn2([Value1], [Value2]) |
| BigMul(Value1, Value2) | Returns an Int64 containing the full product of two specified 32-bit numbers. | BigMul([Amount], [Quantity]) |
| Ceiling(Value) | Returns the smallest integer that is greater than or equal to the given numeric expression. | Ceiling([Value]) |
| Cos(Value) | Returns the cosine of the angle defined in radians. | Cos([Value]) |
| Cosh(Value) | Returns the hyperbolic cosine of the angle defined in radians. | Cosh([Value]) |
| Exp(Value) | Returns the exponential value of the given float expression. | Exp([Value]) |
| Floor(Value) | Returns the largest integer less than or equal to the given numeric expression. | Floor([Value]) |
| Log(Value) | Returns the natural logarithm of a specified number. | Log([Value]) |
| Log(Value, Base) | Returns the logarithm of a specified number in a specified Base. | Log([Value], 2) |
| Log10(Value) | Returns the base 10 logarithm of a specified number. | Log10([Value]) |
| Power(Value, Power) | Returns a specified number raised to a specified power. | Power([Value], 3) |
| Rnd() | Returns a random number that is less than 1, but greater than or equal to zero. | Rnd()*100 |
| Round(Value) | Rounds the given value to the nearest integer. | Round([Value]) |
| Sign(Value) | Returns the positive (+1), zero (0), or negative (-1) sign of the given expression. | Sign([Value]) |
| Sin(Value) | Returns the sine of the angle, defined in radians. | Sin([Value]) |

| | | |
|---|---|---|
| Sinh(Value) | Returns the hyperbolic sine of the angle defined in radians. | Sinh([Value]) |
| Sqr(Value) | Returns the square root of a given number. | Sqr([Value]) |
| Tan(Value) | Returns the tangent of the angle defined in radians. | Tan([Value]) |
| Tanh(Value) | Returns the hyperbolic tangent of the angle defined in radians. | Tanh([Value]) |

## String Functions

| Function | Description | Example |
|---|---|---|
| Ascii(String) | Returns the ASCII code value of the leftmost character in a character expression. | Ascii('a') |
| Char(Number) | Converts integerASCIICode to a character. | Char(65) + Char(51) |
| CharIndex(String1, String2) | Returns the starting position of String1 within String2, beginning from the zero character position to the end of a string. | CharIndex('e', 'devexpress') |
| CharIndex(String1, String2, StartLocation) | Returns the starting position of String1 within String2, beginning from the StartLocation character position to the end of a string. | CharIndex('e', 'devexpress', 2) |
| Concat(String1, ... , StringN) | Returns a string value containing the concatenation of the current string with any additional strings. | Concat('A', ')', [ProductName]) |
| Insert(String1, StartPosition, String2) | Inserts String2 into String1 at the position specified by StartPositon | Insert([Name], 0, 'ABC-') |
| Len(Value) | Returns an integer containing either the number of characters in a string or the nominal number of bytes required to store a variable. | Len([Description]) |
| Lower(String) | Returns the String in lowercase. | Lower([ProductName]) |
| PadLeft(String, Length) | Left-aligns characters in the defined string, padding its left side with white space characters up to a specified total length. | |
| PadLeft(String, Length, Char) | Left-aligns characters in the defined string, padding its left side with the specified Char up to a specified total length. | PadLeft([Name], 30, '<') |

| | | |
|---|---|---|
| PadRight(String, Length) | Right-aligns characters in the defined string, padding its left side with white space characters up to a specified total length. | PadRight([Name], 30) |
| PadRight(String, Length, Char) | Right-aligns characters in the defined string, padding its left side with the specified Char up to a specified total length. | PadRight([Name], 30, '>') |
| Remove(String, StartPosition, Length) | Deletes a specified number of characters from this instance, beginning at a specified position. | Remove([Name], 0, 3) |
| Replace(String, SubString2, String3) | Returns a copy of String1, in which SubString2 has been replaced with String3. | Replace([Name], 'The ', '' |
| Reverse(String) | Reverses the order of elements within a string. | Reverse([Name]) |
| Substring(String, StartPosition, Length) | Retrieves a substring from String. The substring starts at StartPosition and has the specified Length.. | Substring([Description], 2, 3) |
| Substring(String, StartPosition) | Retrieves a substring from String. The substring starts at StartPosition. | Substring([Description], 2) |
| ToStr(Value) | Returns a string representation of an object. | ToStr([ID]) |
| Trim(String) | Removes all leading and trailing SPACE characters from String. | Trim([ProductName]) |
| Upper(String) | Returns String in uppercase. | Upper([ProductName]) |

# Constants

| Constant | Description | Description |
|---|---|---|
| String constants | String constants must be wrapped in apostrophes. | [Country] == 'France' |
| Date-time constants | Date-time constants must be wrapped in '#'. | [OrderDate] >= #1/1/2009# |
| True | Represents the Boolean True value. | [InStock] == True |
| False | Represents the Boolean False value. | [InStock] == False |
| ? | Represents a null reference, one that does not refer to any object. | [Region] != ? |