

Here is a list of the functions that are available in the Extender and SmartList Builder Calculations. This includes a definition of what the function does and a few syntax examples for the function.

## Arithmetic Functions

- **(Subtract)** Subtracts one numeric value from another numeric value. The Subtract function can also be used to subtract a given number of days from a date.

**Examples:**

- $\langle \text{Field1} \rangle - \langle \text{Field2} \rangle$
- $\langle \text{Field1} \rangle - 4$

% **(Modulus)** Returns the integer remainder of dividing the first numeric value by the second numeric value. For example,  $15 \% 6$ , would return 3. This is figured by dividing 15 by 6. The remainder would be 3.

**Examples:**

- $\langle \text{Field1} \rangle \% \langle \text{Field2} \rangle$
- $\langle \text{Field1} \rangle \% 4$

\* **(Multiply)** Multiplies two numeric values together.

**Examples:**

- $\langle \text{Field1} \rangle * \langle \text{Field2} \rangle$
- $\langle \text{Field1} \rangle * 4$

/ **(Divide)** Divides one numeric value by another numeric value.

**Examples:**

- $\langle \text{Field1} \rangle / \langle \text{Field2} \rangle$
- $\langle \text{Field1} \rangle / 4$

+ **(Add)** Adds two numeric values. The Add function can also be used to add a given number of days to a date.

**Examples:**

- $\langle \text{Field1} \rangle + \langle \text{Field2} \rangle$
- $\langle \text{Field1} \rangle + 4$

## Date Functions

This table contains a list of the dateparts that can be used in the following date functions.

Datepart	Abbreviation
year	yy, yyyy
quarter	qq, q
month	mm, m
dayofyear	dy, y
day	dd, d
week	wk, ww
weekday	Dw

Datepart	Abbreviation
hour	hh
minute	mi, n
second	ss, s
millisecond	ms

**DATEADD** Returns a new datetime value that is based on adding an interval to the specified date. The format of the DATEADD function is DATEADD (datepart, number, date). Datepart refers to the datepart from the chart above, number refers to the value to add to the date specified, and date refers to the data to use for the calculation.

**Syntax:**

- DATEADD (datepart, number, date)

**Examples:**

- DATEADD (day, 4, <Field1>)
- DATEADD (month, 4, GETDATE())

**DATEDIFF** Returns the number of date and time boundaries that exist between two specified dates. The format of the DATEDIFF function is DATEDIFF (datepart, startdate, enddate). Datepart refers to the datepart from the chart above, startdate refers to the starting date, and enddate refers to the end date to use in finding the difference.

**Syntax:**

- DATEDIFF (datepart, startdate, enddate)

**Examples:**

- DATEDIFF (year, <Field1>, <Field2>)
- DATEDIFF (month, <Field1>, GetDate())

**DATENAME** Returns a character string that represents the requested part of the date specified. For example, getting the month from the date of 03/10/13 would return March. The format of the DATENAME function is DATENAME (datepart, date). Datepart refers to the datepart from the chart above and the date refers to the date given that the name is to be calculated for.

**Syntax:**

- DATENAME (datepart, date)

**Examples:**

- DATENAME (year, <Field1>)
- DATENAME (month, GetDate())

**DATEPART** Returns an integer that represents the specified part of the specified date. The format of the DATEPART function is DATEPART (datepart, date). Datepart refers to the datepart from the chart above and the date refers to the date given that the part is to be calculated for.

**Syntax:**

- DATEPART (datepart, date)

**Examples:**

- DATEPART (year, <Field1>)
- DATEPART (month, GetDate())

**DAY** Returns an integer that represents the day part of the specified date. This would be the same as using the DATEPART (day, date) function. The format of the DAY function is DAY (date). Date refers to the date given that day is to be calculated for.

**Syntax:**

- DAY (date)

**Examples:**

- DAY (<Field1>)
- DAY (GetDate())

**GETDATE** Returns the current system date and time from the SQL server that hosts the Microsoft Dynamics GP data. This function is not a fixed date, so if using this function one day, it will return a different value than the next.

**Syntax:**

- GETDATE()

**MONTH** Returns an integer that represents the month part of the specified date. This would be the same as using the DATEPART (month, date) function. The format of the MONTH function is MONTH (date). Date refers to the date given that day is to be calculated for.

**Syntax:**

- MONTH (date)

**Examples:**

- MONTH (<Field1>)
- MONTH (GetDate())

**YEAR** Returns an integer that represents the year part of the specified date. This would be the same as using the DATEPART (year, date) function. The format of the YEAR function is YEAR (date). Date refers to the date given that day is to be calculated for.

**Syntax:**

- YEAR (date)

**Examples:**

- YEAR (<Field1>)
- YEAR (GetDate())

## Mathematical Functions

**ABS** Returns the absolute positive value of a given numeric value.

**Syntax:**

- ABS (numeric\_expression)

**Examples:**

- ABS(<Field1>)
- ABS (-6.42)

**CEILING** Returns the smallest integer that is greater than or equal to the given numeric value.

**Syntax:**

- CEILING (numeric\_expression)

**Examples:**

- CEILING (<Field1>)
- CEILING (6.42)

**COS** Returns the trigonometric cosine of an angle in the given numeric value.

**Syntax:**

- COS (numeric\_expression)

**Examples:**

- COS (<Field1>)
- COS (6)

**EXP** Returns the natural exponential of the specified float value.

**Syntax:**

- EXP (numeric\_expression)

**Examples:**

- EXP (<Field1>)
- EXP (6.42)

**FLOOR** Returns the largest integer that is less than or equal to the specified numeric value.

**Syntax:**

- FLOOR (numeric\_expression)

**Examples:**

- FLOOR (<Field1>)
- FLOOR (6.42)

**LOG** Returns the natural (base-2) logarithm for the specified float value.

**Syntax:**

- LOG (numeric\_expression)

**Examples:**

- LOG (<Field1>)
- LOG (6.42)

**LOG10** Returns the base-10 logarithm for the specified float value.

**Syntax:**

- LOG10 (numeric\_expression)

**Examples:**

- LOG10 (<Field1>)
- LOG10 (6.42)

**PI** Returns the value of PI (3.14159265358979...).

**Syntax:**

- PI()

**POWER** Returns the value of the numeric expression raised to the specified power. The format of POWER is POWER(numeric\_expression, y) where numeric\_expression is the value to raise and y is the power to raise it to.

**Syntax:**

- POWER (numeric\_expression , y)

**Examples:**

- POWER (<Field1>, 4)
- POWER (<Field1>, <Field2>)

**RAND** Returns a random float value between 0 and 1.

**Syntax:**

- RAND()

**ROUND** Returns a numeric value, rounded to the specified length or precision. The format for ROUND is ROUND (numeric\_expression, length) where numeric\_expression is the value to be rounded and length is the number of places to round the value to.

**Syntax:**

- ROUND (numeric\_expression, length)

**Examples:**

- ROUND (<Field1>, 2)
- ROUND (6.42642, 2)

**SIGN** Returns the positive sign (+1), the zero sign (0), or the negative sign (-1) of the specified numeric value.

**Syntax:**

- SIGN (numeric\_expression)

**Examples:**

- SIGN (<Field1>)
- SIGN (-64)

**SIN** Returns the trigonometric sine of an angle in the given numeric value.

**Syntax:**

- SIN (numeric\_expression)

**Examples:**

- SIN (<Field1>)
- SIN (6)

**SQRT** Returns the square root of the given numeric value.

**Syntax:**

- SQRT (numeric\_expression)

**Examples:**

- SQRT (<Field1>)
- SQRT (642)

**SQUARE** Returns the square of the given numeric value.

**Syntax:**

- SQUARE (numeric\_expression)

**Examples:**

- SQUARE (<Field1>)
- SQUARE (642)

**TAN** Returns the tangent of an angle in the given numeric value.

**Syntax:**

- TAN (numeric\_expression)

**Examples:**

- TAN (<Field1>)
- TAN (6)

## String functions

**ASCII** Returns the ASCII code value of the leftmost character of a character string.

**Syntax:**

- ASCII (character\_expression)

**Examples:**

- ASCII (<Field1>)
- ASCII (J)

**CHAR** Converts a given integer ASCII code to a character. The integer\_expression value accepts an integer from 0 through 255. NULL is returned if the integer expression is not in this range.

**Syntax:**

- CHAR (integer\_expression)

**Examples:**

- CHAR (<Field1>)
- CHAR (20)

**LEFT** Returns the left most characters of a string up to the length of the value given. If the length given equals 0, it returns an empty string. If length given is equal to or greater than the length of the string, the entire string is returned.

**Syntax:**

- LEFT (character\_expression , integer\_expression)

**Examples:**

- LEFT (<Field1>, 4)
- LEFT ('string of text', 4)

**LEN** Returns the length of the given character string.

**Syntax:**

- LEN (string\_expression)

**Examples:**

- LEN (<Field1>)
- LEN ('string of text')

**LOWER** Returns a string of characters all converted to lowercase.

**Syntax:**

- LOWER (character\_expression)

**Examples:**

- LOWER (<Field1>)
- LOWER ('UPPERCASE')

**LTRIM** Removes leading blank spaces from a character string and then returns the modified character string.

**Syntax:**

- LTRIM (character\_expression)

**Examples:**

- LTRIM (<Field1>)
- LTRIM (' string of text')

**REPLACE** Returns a character string after it replaces all occurrences of a string expression with a new string expression. The format of the REPLACE function is REPLACE (target\_string, string\_to\_replace, replacement\_string). Target\_string is the string expression to be searched, string\_to\_replace is the substring to find and replace, and replacement\_string is the string to replace the substring with.

**Syntax:**

- REPLACE (target\_string , string\_to\_replace , replacement\_string)

**Examples:**

- REPLACE (<Field1>, <Field2>, <Field3>)
- REPLACE (<Field1>, 'description', 'new description')

**REPLICATE** Returns a character string repeated a specified number of times.

**Syntax:**

- REPLICATE (character\_expression , integer\_expression)

**Examples:**

- REPLICATE (<Field1>, <Field2>)
- REPLICATE (<Field1>, 4)

**REVERSE** Returns the reverse of a character string.

**Syntax:**

- REVERSE (character\_expression)

**Examples:**

- REVERSE (<Field1>)
- REVERSE ('Backwards')

**RIGHT** Returns the last substring characters of a string to the length specified. If the length specified equals 0, the **RIGHT** function returns an empty string. If the length is equal to or greater than the length of the string, the function returns the complete string.

**Syntax:**

- RIGHT (character\_expression , integer\_expression)

**Examples:**

- RIGHT (<Field1>, 4)
- RIGHT ('string of text', 4)



**RTRIM** Removes all the trailing blank spaces in a character string and then returns the modified character string.

**Syntax:**

- RTRIM (character\_expression)

**Examples:**

- RTRIM (<Field1>)
- RTRIM ('string of text ')

**SPACE** Returns a string of repeated spaces to the length specified in the integer value.

**Syntax:**

- SPACE (integer\_expression)

**Examples:**

- SPACE (<Field1>)
- SPACE (6)

**STR** Converts a numeric expression to a string expression.

**Syntax:**

- STR (numeric\_expression)

**Examples:**

- STR (<Field1>)
- STR (642.46)

**STUFF** Replaces a specified length of characters with another set of characters at a specified starting point. The format for the STUFF function is STUFF (character\_expression, start, length, stuff\_characters). The character\_expression is the string to evaluate, the start is the position to start the replacement at, the length is the number of characters to remove, and the stuff\_characters is the string to replace those removed with.

**Syntax:**

- STUFF (character\_expression , start , length , stuff\_characters)

**Examples:**

- STUFF (<Field1>, 2, 3, <Field2>)
- STUFF ('This is a string of text', 1, 9, 'The')

**SUBSTRING** Returns part of a character expression. The format for the SUBSTRING function is SUBSTRING (expression, start, length). The expression is the character string to evaluate, start is the integer value specifying the starting position in the expression, and the length is an integer value of how many characters to return.

**Syntax:**

- SUBSTRING (expression, start, length)

**Examples:**

- SUBSTRING (<Field1>, <Field2>, 4)
- SUBSTRING ('string of text', 2, 4)

**UPPER** Returns a string of characters all converted to uppercase.

**Syntax:**

- UPPER (character\_expression)

**Examples:**

- UPPER (<Field1>)
- UPPER ('lowercase')

## System functions

**CASE** Evaluates several conditions and then return a single value for each condition.

**Syntax:**

- CASE input\_expression  
WHEN when\_expression THEN result\_expression  
ELSE else\_result\_expression  
END

**Examples:**

- CASE <Field1>  
WHEN value1 THEN <Field1> + <Field2>  
WHEN value2 THEN <Field1> + <Field3>  
ELSE <Field1> + <Field4>  
END
- CASE <Field1>  
WHEN YEAR (GetDate()) THEN Current  
ELSE History  
END

**CAST** Converts expressions of one data type to another data type. The format of the CAST function is CAST (expression AS data\_type). Expression is the data to be converted and datatype is the data type to convert the expression to.

**Syntax:**

- CAST (expression AS data\_type)

**Examples:**

- CAST (<Field1> AS int)
- CAST (20 AS string)