

Task Factory Functions List K-L

Last Modified on 02 October 2020

🔔 Task Factory users running version 2020.1.4 or older (released prior to May 27, 2020): **There's an important Task Factory update.** Please visit [here](#) for more details.

Last

Description			Aggregation that returns the last value of the group. Can only be used as part of the Advanced Aggregation component
Syntax			Last(Value [, Condition])
Returns			Object.
Parameters			
Name	Optional	Description	
Value	false	The value to add to the aggregation	
Condition	false	The condition allows you to conditionally add values to the aggregation based on the expression. For instance, you could define the aggregate as Last(SalesTotal, CustomerState="FL") and this would only add the value to the aggregation if the customers state was florida.	
Examples			
Last(SalesTotal), Last(SalesTotal, CustomerState="FL")			

LastDate

Description		Returns the last day of a specified month.
Syntax		LastDate(), LastDate(Date)

Returns	returns date	
LastDate		
Parameters		
Name	Optional	Description
Date	false	Date to retrieve value from
Examples		
LastDate(), LastDate("12/8/2011"), LastDate(varDate), LastDate(OrderDateColumn)		

LastDateOfPrevMonth		
Description	Returns the last day of the previous month.	
Syntax	LastDateOfPrevMonth(), LastDateOfPrevMonth(Date)	
Returns	returns date	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve value from
Examples		
LastDateOfPrevMonth(), LastDateOfPrevMonth("12/8/2011"), LastDateOfPrevMonth(varDate), LastDateOfPrevMonth(OrderDateColumn)		

Least		
Description	Returns the smallest value from a list of input values. By default, the match is not case sensitive.	
Syntax	LEAST(Case_Flag, Value1 [, Value2, ..., ValueN,])	
Returns	value1 if it is the smallest of the input values, value2 if it is the smallest of the input values, and so on. NULL if any of the arguments is null.	

Parameters		
Name	Optional	Description
Least		
CaseFlag	false	Determines whether the arguments in this function are case sensitive. You can enter any valid transformation expression. When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.
Value	false	Any datatype except Binary. Datatype must be compatible with other values. Value you want to compare against other values. You must enter at least one value argument. If the value is Numeric, and other input values are of other numeric datatypes, all values use the highest precision possible. For example, if some values are of the Integer datatype and others are of the Double datatype, the Integration Service converts the values to Double.
Examples		
Least(false, varValues), Least(false, "test1", "test2", "test3")		

Left		
Description	Returns a string containing a specified number of characters from the left side of a string..	
Syntax	Left(String, Number_Of_Characters)	
Returns	String. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String		String datatype. String expression from which the

	false	leftmost characters are returned.
Number_Of_Characters	false	The number of characters to return from the string.
Examples		
Left("Stop Here. None of this will be there", 10), Left(FirstNameColumn, 20)		

<h2>Length</h2>		
Description	Returns the number of characters in a string, including trailing blanks.	
Syntax	Length(String)	
Returns	Integer representing the length of the string. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	String datatype. The strings you want to evaluate. You can enter any valid transformation expression.
Examples		
Length("How Long Is This?"), Length(FirstNameColumn)		

<h2>Log</h2>		
Description	Returns the logarithm of a numeric value. Most often, you use this function to analyze scientific data rather than business data.	
Syntax	LOG(Base, Exponent)	
Returns	Double value. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description

Base	false	The base of the logarithm. Must be a positive numeric value other than 0 or 1. Any valid task editor that evaluates to a positive number other than 0 or 1.
Exponent	false	The exponent of the logarithm. Must be a positive numeric value greater than 0. Any valid task editor that evaluates to a positive number greater than 0.
Examples		
Log(BaseColumn, ExponentColumn)		

<h2>Log10</h2>		
Description	Returns the base 10 logarithm of a specified number.	
Syntax	Log10(Numeric_Value)	
Returns	Numeric	
Parameters		
Name	Optional	Description
Numeric_Value	false	Passes the values for which you want to the Atan of
Examples		
Log10(56)		

<h2>LogBufferRowToFile</h2>	
Description	Logs a row of data from a dataflow to a flat file.
Syntax	LogBufferRowToFile(File_Path [, Delimiter] [, Quoted_String])

Returns	Nothing	
LogBufferRowToFile		
Name	Optional	Description
File_Path	false	The path to the file to log the buffer row to.
Delimiter	optional	The file delimiter to be used to separate the column values. Default is ","
Quoted_String	optional	Tells the engine whether to wrap string values in quotes or not. Default is false.
Examples		
LogBufferRowToFile("C:\temp\logfile.ext"), LogBufferRowToFile("c:\temp\logfile.txt", " ", True)]		

LogError		
Description	Logs an error message to the ssis output log (progress tab in BIDS)	
Syntax	LogError(Message [, Condition])	
Returns	Nothing	
Parameters		
Name	Optional	Description
Message	false	The message to log.
Condition	true	The condition to evaluate to true before logging the message
Examples		
LogError("This is a error"), LogError("Values didn't match", value1!=value2)]		

LogInfo

Description	Logs an informational message to the ssis output log (progress tab in BIDS)	
Syntax	LogInfo(Message [, Condition])	
Returns	Nothing	
Parameters		
Name	Optional	Description
Message	false	The message to log.
Condition	true	The condition to evaluate to true before logging the message
Examples		
LogInfo("This is a message"), LogInfo("Values didn't match", value1!=value2)]		

LogWarning

Description	Logs a warning message to the ssis output log (progress tab in BIDS)	
Syntax	LogWarning(Message [, Condition])	
Returns	Nothing	
Parameters		
Name	Optional	Description
Message	false	The message to log.
Condition	true	The condition to evaluate to true before logging the message
Examples		

```
LogWarning("This is a warning"), LogWarning("Values didn't match", value1!=value2)]
```

LogWarning

LookupData

Description	LookupData will use an Advanced Lookup Cache Connection Manager to retrieve data from the cached connection manager. LookupData will return the value of the first output column defined on a cache connection manager. The args parameter array is used to pass in values that match the "Input Parameters" defined on a cache connection manager.	
Syntax	LookupData(Cache_Connection_Manager, Arg1 [, Arg2, ..., ArgN])	
Returns	returns the value of the output column if a match is found. Returns NULL if no match is found.	
Parameters		
Name	Optional	Description
Cache_Connection_Manager	false	The Advanced Lookup Cache Connection Manager where the lookup will be performed. Must use the @@[ConnectionManager] syntax.
Args	false	Args is a parameter array of values that directly correlate to the input parameters defined on a cache connection manager.
Examples		
<p>LookupData(@@[ProductCache], ProductID) - In this example, the cache connection manager (@@[ProductCache]) is storing the products. ProductID is a column that is part of the current component being used (Advanced Derived Column, Advanced Aggregator, etc) and the value of ProductID is passed into LookupData from the Data Flow.</p> <p>[LookupData(@@[ProductCache], ProductID, OrderDate)] - In this example, the cache connection manager (@@[ProductCache]) is storing the products. ProductID and OrderDate are columns that are part of the current component being used (Advanced Derived Column, Advanced Aggregator, etc) and the value of them is passed into LookupData from the Data Flow.</p>		

LookupDataByColumn

LookupDataByColumn will use an Advanced Lookup Cache Connection Manager to retrieve data from the cached connection

LookupDataByColumn

Description	manager and return a specified. Unlike LookupData, which returns the value of the first output column defined on a cache connection manager, LookupDataByColumn can be used to grab a specific column. The Arg parameter array is used to pass in values that match the "Input Parameters" defined on a cache connection manager.	
Syntax	LookupDataByColumn(Cache_Connection_Manager, Column_To_Retrieve, Arg1 [, Arg2, ..., ArgN])	
Returns	returns the value of the output column if a match is found. Returns NULL if no match is found.	
Parameters		
Name	Optional	Description
Cache_Connection_Manager	false	The Advanced Lookup Cache Connection Manager where the lookup will be performed. Must use the @@[ConnectionManager] syntax.
Column_To_Retrieve	false	The name of an column defined as "Output" on the Advanced Lookup Cache Connection Manager.
Args	false	Args is a parameter array of values that directly correlate to the input parameters defined on a cache connection manager.
Examples		
<p>LookupData(@@[ProductCache], "ProductName", ProductID) - In this example, the cache connection manager (@@[ProductCache]) is storing the products. ProductID is a column that is part of the current component being used (Advanced Derived Column, Advanced Aggregator, etc) and the value of ProductID is passed into LookupData from the Data Flow. "ProductName" is the column being retrieved from the cache connection manager.</p> <p>[LookupData(@@[ProductCache], "ProductCost", ProductID, OrderDate)] - In this example, the cache connection manager (@@[ProductCache]) is storing the products. ProductID and OrderDate are columns that are part of the current component being used (Advanced Derived Column, Advanced Aggregator, etc) and the value of them is passed into LookupData from the Data Flow. "ProductCost" is the column being retrieved from the cache connection manager.</p>		

Lower

Description	Converts uppercase string characters to lowercase.
-------------	--

Syntax	LOWER(String)	
Returns	Lowercase character string. If the data contains multibyte characters, the return value depends on the code page and data movement mode of the Integration Service. NULL if a value in the selected port is NULL.	
Parameters		
Name	Optional	Description
String	false	Any string value. The argument passes the string values that you want to return as lowercase. You can enter any valid transformation expression that evaluates to a string.
Examples		
Lower("Lower All THIS"), Lower(LastNameColumn)		

<h2>LPad</h2>		
Description	Adds a set of blanks or characters to the beginning of a string to set the string to a specified length.	
Syntax	LPad(First_String, Length [, Second_String])	
Returns	String of the specified length. NULL if a value passed to the function is NULL or if length is a negative number.	
Parameters		
Name	Optional	Description
First_String	false	Can be a character string. The strings you want to change. You can enter any valid task editor.
Length	false	Must be a positive integer literal. This argument specifies the length you want each string to be.
		Can be any string value. The characters you want to append to the left-side of the

Second_String	true	LPad first_string values. You can enter any valid task editor. You can enter a specific string literal. However, enclose the characters you want to add to the beginning of the string within single quotation marks, as in 'abc'. This argument is case sensitive. If you omit the second_string, the function pads the beginning of the first string with blanks.
Examples		
LPad("SentryOne", 30), LPad(CompanyNameColumn, 50)		

LTrim		
Description	Removes blanks or characters from the beginning of a string.	
Syntax	LTRIM(String [, Trim_Set])	
Returns	String. The string values with the specified characters in the trim_set argument removed. NULL if a value passed to the function is NULL. If the trim_set is NULL, the function returns NULL.	
Parameters		
Name	Optional	Description
String	false	Any string value. Passes the strings you want to modify. You can enter any valid task editor. Use operators to perform comparisons or concatenate strings before removing characters from the beginning of a string.
Trim_Set	true	Any string value. Passes the characters you want to remove from the beginning of the first string. You can enter any valid task editor. You can also enter a character string. However, you must enclose the characters you want to remove from the beginning of the string within single quotation marks, for example, 'abc'. If you omit the second

LTrim

string, the function removes any blanks from the beginning of the string. LTRIM is case sensitive. For example, if you want to remove the 'A' character from the string 'Alfredo', you would enter 'A', not 'a'.

Examples

```
LTrim(" SentryOne"), LTrim(CompanyNameColumn), LTrim("----SentryOne", "-")
```