

# Task Factory Functions List I-J

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.

## IfNull

Description	Tests for a null value and, if it exists, returns a default value.	
Syntax	IfNull( Value, defaultValue )	
Returns	defaultValue	
Parameters		
Name	Optional	Description
Value	false	The column/variable/string to test for NULL value(s).
defaultValue	false	The value returned when NULL is present
Examples		
IfNull(PhoneNumber, 'Unknown'), IfNull(IntRate, @[User::dailyRate])		

## IfNullOrEmpty

Description	Tests for a null or empty value and, if it exists, returns a default value.	
Syntax	IfNullOrEmpty( Value, defaultValue )	
Returns	defaultValue	
Parameters		
Name	Optional	Description

Value	false <b>IfNullOrEmpty</b>	The column/variable/string to test for NULL or empty value(s).
defaultValue	false	The value returned when NULL is present
Examples		
IfNullOrEmpty(PhoneNumber, 'Unknown'), IfNullOrEmpty(IntRate, @[User::dailyRate])		

<b>IIF</b>		
Description	Returns one of two values you specify, based on the results of a condition.	
Syntax	IIF( Condition, Value1 , Value2 )	
Returns	value1 if the condition is TRUE. value2 if the condition is FALSE.	
Parameters		
Name	Optional	Description
Condition	false	The condition you want to evaluate. You can enter any valid task editor that evaluates to TRUE or FALSE.
Value1	false	Any datatype except Binary. The value you want to return if the condition is TRUE. The return value is always the datatype specified by this argument. You can enter any valid task editor, including another IIF expression.
Value2	true	Any datatype except Binary. The value you want to return if the condition is FALSE. You can enter any valid task editor, including another IIF expression.
Examples		
IIF( SALES > 100, EMP_NAME )		

SALES	EMP_NAME	RETURN VALUE
150	John Smith IIF	John Smith
50	Pierre Bleu	" (empty string)
120	Sally Green	Sally Green
NULL	Greg Jones	" (empty string)

IN		
Description	Searches value in the given list. If value is found in the list then it will return True else False. You can perform case-sensitive or case-insensitive compare based on last (optional) argument.	
Syntax	IN( Value_To_Find, value1 [, value2, value3..., valueN] [, caseFlag])	
Returns	Boolean - True if the value is found, False if the value is not found	
Parameters		
Name	Optional	Description
Value_To_Find	false	The value you want to search for in the array.
List_Of_Values	false	List of values to be compared.
CaseFlag	true	(Optional) This flag specifies how to compare string data. If this flag is 0 then it will do case-insensitive compare. Anything other than 0 will be case-sensitive compare. By default it will do case-sensitive compare means "DEC" is not same as "Dec"
Examples		

In(MONTHNAME("12/12/2013",True), "JAN",FEB", "DEC") --This should return False. Last argument not specified means case-sensitive compare. In(MONTHNAME("12/12/2013",True), "JAN",FEB", "DEC",0) -- This should return True. Last argument=0 means case-insensitive compare. In(MONTHNAME("12/12/2013",True), "JAN",FEB", "DEC",1) --This should return False. Last argument=1 means case-insensitive compare.

## InitCap

Description	Capitalizes the first letter in each word of a string and converts all other letters to lowercase.	
Syntax	InitCap( String )	
Returns	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 passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	Any datatype except Binary. You can enter any valid task editor.
Value1	false	para2 desc.
Examples		
InitCap("a fox runs over the river"), InitCap(FirstNameColumn), InitCap(FirstNameColumn + " " + LastNameColumn)		

## InsertChars

Description	Inserts characters into a string at a defined index	
Syntax	InsertChars( Base_String, Index, String_To_Insert )	
Returns	String. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description

Base_String	false	The base string in which characters will be inserted.
Index	<b>InsertChars</b> false	The index of the Base_String where String_To_Insert will be inserted
String_To_Insert	false	The string to insert at the index.
Examples		
InsertChar("400 Dr", 4, "College ") will return "400 College Dr", InsertChar(FullNameColumn, InStr(FullName, " "), "-")		

<h2>InStr</h2>		
Description	Returns the position of a character set in a string, counting from left to right.	
Syntax	INSTR( String_To_Search, Search_Value [, Start] [, Occurrence] [, Comparison_Type ] )	
Returns	Integer if the search is successful. Integer represents the position of the first character in the search_value, counting from left to right. 0 if the search is unsuccessful. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String_To_Search	false	The string must be a character string. Passes the value you want to evaluate. You can enter any valid transformation expression. The results of the expression must be a character string. If not, INSTR converts the value to a string before evaluating it.
Search_Value	false	Any value. The search value is case sensitive. The set of characters you want to search for. The Search_Value must match a part of the string. For example, if you write INSTR('Alfred Pope', 'Alfred Smith') the function returns 0. You can enter any valid

	<b>InStr</b>	transformation expression. If you want to search for a character string, enclose the characters you want to search for in single quotation marks, for example 'abc'.
Start	true	Must be an integer value. The position in the string where you want to start the search. You can enter any valid transformation expression. The default is 1, meaning that INSTR starts the search at the first character in the string. If the start position is 0, INSTR searches from the first character in the string. If the start position is a positive number, INSTR locates the start position by counting from the beginning of the string. If the start position is a negative number, INSTR locates the start position by counting from the end of the string. If you omit this argument, the function uses the default value of 1.
Occurrence	true	A positive integer greater than 0. You can enter any valid transformation expression. If the search value appears more than once in the string, you can specify which occurrence you want to search for. For example, you would enter 2 to search for the second occurrence from the start position. If you omit this argument, the function uses the default value of 1, meaning that INSTR searches for the first occurrence of the search value. If you pass a decimal, the Integration Service rounds it to the nearest integer value. If you pass a negative integer or 0, the dataflow fails.
		The string comparison type, either linguistic or binary, when the Integration Service runs in Unicode mode. When the Integration Service runs in ASCII mode, the comparison type is always binary.

Comparison_Type	true	<b>InStr</b> Linguistic comparisons take language-specific collation rules into account, while binary comparisons perform bitwise matching. For example, the German sharp s character matches the string "ss" in a linguistic comparison, but not in a binary comparison. Binary comparisons run faster than linguistic comparisons. Must be an integer value, either 0 or 1: - 0: INSTR performs a linguistic string comparison. - 1: INSTR performs a binary string comparison. Default is 0.
Examples		
INSTR("Is four in this string", "four"), InStr(ProductDescriptionColumn, "Blue"), InStr(ProductDescriptionColumn, "Blue", 0, 2, 1)		

<b>IsDate</b>		
Description	Determines whether the value is a date.	
Syntax	IsDate( Value )	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsDate("7/7/2012")		

<b>IsEmpty</b>	
Description	Determines whether the value is empty.
Syntax	IsEmpty( Value )

Returns	Boolean	
<b>IsEmpty</b>		
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsEmpty(AddressColumn)		

<b>IsNull</b>		
Description	Determines whether the value is null.	
Syntax	IsNull( Value )	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsNull(AddressColumn)		

<b>IsNullOrEmpty</b>		
Description	Determines whether the value is null or empty.	
Syntax	IsNullOrEmpty( Value )	
Returns	Boolean	
Parameters		



Name	Optional	Description
Value	false	The value to check.
Examples		
IsNullOrEmpty(AddressColumn)		

IsNumber		
Description	Determines whether the value is a number.	
Syntax	IsNumber( Value )	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		
IsNumber("12345")		

IsSpaces		
Description	Determines whether the value is all spaces.	
Syntax	IsSpaces( Value )	
Returns	Boolean	
Parameters		
Name	Optional	Description
Value	false	The value to check.
Examples		

```
IsSpaces("test")
```

# IsSpaces