

# Task Factory Functions List R

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.

## Rand

Description	Returns a random number between 0 and 1. This is useful for probability scenarios.	
Syntax	RAND(), RAND( Seed ), RAND( Min, Max)	
Returns	Numeric. For the same seed, the Integration Service generates the same sequence of numbers.	
Parameters		
Name	Optional	Description
Seed	true	Double. Starting value for the Integration Service to generate the random number. Value must be a constant. If you do not enter a seed, the Integration Service uses the current system time to derive the numbers of seconds since January 1, 1971. It uses this value as the seed.
Min	true	Integer. Minimum value for the generate the random number. Value must be a constant.
Max	true	Integer. Maximum value for the generate the random number. Value must be a constant.
Examples		
Rand(), Rand(.5), Rand(1, 10)		

## RegexExtract

Description	Extract values out of a string based on a regular expression	
Syntax	RegexExtract( Input_String, Pattern [, Instance_Of_Pattern_To_Extract] )	
Returns	String.	
Parameters		
Name	Optional	Description
Input_String	false	String value. The string that will be manipulated by the regular expression replace.
Pattern	false	String value. Regular expression pattern.
Instance_Of_Pattern_To_Extract	true	Integer. If more than one instance of the pattern is found in input_string, the value returned will be based on the value provided.
Examples		
RegexExtract("Get numbers 1234", "[0-9]+"), RegexExtract("Get second set of numbers 1234 4567", "[0-9]+", 2)		

## RegexMatch

Description	Performs a regular expression match against a string	
Syntax	RegexMatch( Input_String, Pattern [, Ignore_Case] )	
Returns	Boolean. Null if input_string or pattern parameters are null.	
Parameters		
Name	Optional	Description
Input_String	false	String value. The string that will be searched and matched.

Pattern	false	String value. Regular expression pattern.
Ignore_Case	true	Boolean. Specifies whether the match will be case sensitive.
<b>RegMatch</b>		
Examples		
RegMatch("123434-", "^[0-9]+\$"), RegMatch("Sentry One", "[a-zA-Z\s]+\$")		

<b>RegReplace</b>		
Description	Replace values in a string based on a regular expression	
Syntax	RegReplace( Input_String, Pattern, Replacement [, Number_Of_Replacements] )	
Returns	String.	
Parameters		
Name	Optional	Description
Input_String	false	String value. The string that will be manipulated by the regular expression replace.
Pattern	false	String value. Regular expression pattern.
Replacement	false	String value. Replacement value if the pattern is found.
Number_Of_Replacements	true	Integer. The number of times the pattern should be replaced.
Examples		
RegReplace("Replace numbers 1234", "[0-9]+", ""), RegReplace("Replace first set of numbers 1234 4567", "[0-9]+", "", 1)		

<b>RegToken</b>	

Description	Splits a string based on a regular expression.	
Syntax	RegToken( Input_String, Delimiter_Regex, Occurrence [, Ignore_Case] )	
Returns	String. If the input_string or delimiter_regex is NULL, the function returns NULL.	
Parameters		
Name	Optional	Description
Input_String	false	A string value that will be split by the delimiter_regex parameter.
Delimiter_Regex	false	A regular expression to be used to split the value of Input_String.
Occurrence	false	The occurrence that will be returned after the string has been split.
Ignore_Case	true	Tells the regular expression engine whether or not to perform the split using case sensitivity.
Examples		
RegToken("Sentry~44~One~55~Inc", "[0-9]+~", 2, false)		

## RegTokenCount

Description	Returns the nksn in a string that contains tokens separated by the regular expression.	
Syntax	RegTokenCount( Input_String, Expression_Regex [, Ignore_Case] )	
Returns	Integer. If the input_string or expression_regex is NULL, the function returns NULL.	
Parameters		
Name	Optional	Description
Input_String	false	A string value that will be searched by the expression_regex.

Expression_Regex	false	A regular expression to be located in the Input_String.
Ignore_Case	true	Tells the regular expression engine whether or not to perform the split using case sensitivity.
Examples		
RegTokenCount("Sentry~44~One~55~Inc", "[0-9]+", false)		

## RemoveChars

Description	Removes a portion of a string starting at an index.	
Syntax	RemoveChars( String, Start [, Length] )	
Returns	String. Empty string if you pass a negative or 0 length value. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	Must be a character string. Passes the string you want to remove characters from. You can enter any valid task editor. If you pass a numeric value, the function converts it to a character string.
Start	false	Must be an integer. The position in the string where you want to start removing characters. You can enter any valid task editor. If the start position is a positive number, RemoveChars locates the start position by counting from the beginning of the string. If the start position is a negative number, RemoveChars locates the start position by counting from the end of the string. If the start position is 0, RemoveChars searches from the first character in the string.

Length	<h2 style="text-align: center;">RemoveChars</h2> <p style="text-align: center;">true</p>	<p>Must be an integer greater than 0. The number of characters you want RemoveChars to return. You can enter any valid task editor. If you omit the length argument, RemoveChars returns all of the characters from the start position to the end of the string. If you pass a negative integer or 0, the function returns an empty string. If you pass a decimal, the function rounds it to the nearest integer value.</p>
Examples		
<p>RemoveChars("This will be cut off here. This won't be here", 27), RemoveChars("This will be cut off here. This will be here too.", 27, 4))</p>		

<h2 style="margin: 0;">ReplaceChr</h2>		
Description	<p>Replaces characters in a string with a single character or no character. REPLACECHR searches the input string for the characters you specify and replaces all occurrences of all characters with the new character you specify.</p>	
Syntax	<p>REPLACECHR( CaseFlag, InputString, OldCharSet, NewChar )</p>	
Returns	<p>String. Empty string if REPLACECHR removes all characters in InputString. NULL if InputString is NULL. InputString if OldCharSet is NULL or empty.</p>	
Parameters		
Name	Optional	Description
CaseFlag	false	<p>Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid task editor. 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.</p>
		<p>Must be a character string. Passes the string you want to search. You can enter any valid</p>

InputString	false	task editor. If you pass a numeric value, the function converts it to a character string. If InputString is NULL, REPLACECHR returns NULL.
<h2>ReplaceChr</h2>		
OldCharSet	false	Must be a character string. The characters you want to replace. You can enter one or more characters. You can enter any valid task editor. You can also enter a text literal enclosed within single quotation marks, for example, 'abc'. If you pass a numeric value, the function converts it to a character string. If OldCharSet is NULL or empty, REPLACECHR returns InputString.
NewChar	false	Must be a character string. You can enter one character, an empty string, or NULL. You can enter any valid task editor. If NewChar is NULL or empty, REPLACECHR removes all occurrences of all characters in OldCharSet in InputString. If NewChar contains more than one character, REPLACECHR uses the first character to replace OldCharSet.
Examples		
REPLACECHR(false, "I can replace this i with t", "i", "t"), REPLACECHR(True, CompanyNameColumn, "_", "_")		

<h2>ReplaceStr</h2>	
Description	Replaces characters in a string with a single character, multiple characters, or no character. REPLACESTR searches the input string for all strings you specify and replaces them with the new string you specify.
Syntax	ReplaceStr ( CaseFlag, InputString, OldString1 [, OldString2, ..., OldStringN], NewString )
Returns	String. Empty string if REPLACESTR removes all characters in InputString. NULL if InputString is NULL. InputString if all OldString arguments are NULL or empty.

Parameters		
Name	Optional	Description
CaseFlag	false	Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid task editor. 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.
InputString	false	Must be a character string. Passes the strings you want to search. You can enter any valid task editor. If you pass a numeric value, the function converts it to a character string. If InputString is NULL, REPLACESTR returns NULL.
OldString	false	Must be a character string. The string you want to replace. You must enter at least one OldString argument. You can enter one or more characters per OldString argument. You can enter any valid task editor. You can also enter a text literal enclosed within single quotation marks, for example, 'abc'. If you pass a numeric value, the function converts it to a character string. When REPLACESTR contains multiple OldString arguments, and one or more OldString arguments is NULL or empty, REPLACESTR ignores the OldString argument. When all OldString arguments are NULL or empty, REPLACESTR returns InputString. The function replaces the characters in the OldString arguments in the order they appear in the function. For example, if you enter multiple OldString arguments, the first OldString argument has precedence over the second OldString argument, and the second OldString argument has precedence over the third OldString argument. When REPLACESTR replaces a string,



	<h2>ReplaceStr</h2>	it places the cursor after the replaced characters in InputString before searching for the next match.
NewString	false	Must be a character string. You can enter one character, an empty string, or NULL. You can enter any valid task editor. If NewChar is NULL or empty, REPLACECHR removes all occurrences of all characters in OldCharSet in InputString. If NewChar contains more than one character, REPLACECHR uses the first character to replace OldCharSet.
Examples		
REPLACESTR(false, "I can replace this with that", "this", "that"), REPLACESTR(True, CompanyAddressColumn, "Street", "St")		

<h2>Reverse</h2>		
Description	Reverses the input string.	
Syntax	Reverse( String )	
Returns	String. Reverse of the input value.	
Parameters		
Name	Optional	Description
String	false	Any character value. Value you want to reverse.
Examples		
Reverse("This will be backwards")		

<h2>Right</h2>	
	Returns a string containing a specified number of characters from

Description	the Right side of a string..	
Syntax	Right( String, Number_Of_Characters )	
Returns	String. NULL if a value passed to the function is NULL.	
Parameters		
Name	Optional	Description
String	false	String datatype. String expression from which the rightmost characters are returned.
Number_Of_Characters	false	The number of characters to return from the string.
Examples		
Right("Only these here", 10), Right(FirstNameColumn, 20)		

## Round(DateTime)

Description	Rounds upto specified part of supplied datetime.	
Syntax	Round( Value [, Format])	
Returns	Date	
Parameters		
Name	Optional	Description
Value	false	A date value to be rounded.
		Date part format which needs to be rounded. See below list for possible format specifier Year = Y, YY, YYYY, YYYY Month = M, MM, MON, MONTH Day = D, DD, DDD, DY, DAY Hour = H, HH, HH12, HH24 Minute = MI Second = S,SS Rules: ===== When format=YY and Month >=7 then Year is increased by one and all other parts after year portion is reset When format=MM and Day >=16

Format	<b>Round(DateTime)</b>	then Month is increased by one and all other parts after month portion is reset When format=DD and Hour >=12 then Day is increased by one and all other parts after day portion is reset When format=HH and Minute >=30 then Hour is increased by one and all other parts after hour portion is reset When format=MI and Second >=30 then Minute is increased by one and all other parts after minute portion is reset When format=SS and Milliseconds >=500 then Second is increased by one and all other parts after second portion is reset
--------	------------------------	--

Examples
<pre> ROUND(TO_DATE('4/16/1998 8:24:19'), 'YY') ==&gt; "1/1/1998" ROUND(TO_DATE('7/16/1998 8:24:19'), 'YY') ==&gt; "1/1/1999" ROUND(TO_DATE('4/15/1998 8:24:19'), 'MM') ==&gt; "4/1/1998" ROUND(TO_DATE('05/22/1998 10:15:29'), 'MM') ==&gt; "6/1/1998" ROUND(TO_DATE('06/13/1998 2:30:45'), 'DD') ==&gt; "6/13/1998" ROUND(TO_DATE('06/13/1998 22:30:45'), 'DD') ==&gt; "6/14/1998" ROUND(TO_DATE('04/01/1998 11:29:35'), 'HH') ==&gt; "04/01/1998 11:00:00" ROUND(TO_DATE('04/01/1998 13:39:00'), 'HH') ==&gt; "04/01/1998 14:00:00" ROUND(TO_DATE('05/22/1998 10:15:29'), 'MI') ==&gt; "05/22/1998 10:15:00" ROUND(TO_DATE('05/22/1998 10:15:30'), 'MI') ==&gt; "05/22/1998 10:16:00" ROUND(TO_DATE('05/22/1998 10:15:29.499'), 'SS') ==&gt; "05/22/1998 10:15:29.000" ROUND(TO_DATE('05/22/1998 10:15:29.500'), 'SS') ==&gt; "05/22/1998 10:15:30.000" </pre>

<h2>Round(Number)</h2>		
Description	Rounds a decimal value to a specified number of fractional digits.	
Syntax	Round( Value [, Decimal_Points] )	
Returns	Decimal	
Parameters		
Name	Optional	Description
Value	false	A decimal number to be rounded.
		The number of decimal places in the return value. If you pass -ve argument then digits

Decimal_Points	true	before decimal point is truncated. If you omit this argument then number is rounded to nearest integer value
<b>Round(Number)</b>		
Examples		
<p>ROUND(12.9938,3) ==&gt; 12.994    ROUND(-18.8679, 3) ==&gt; -18.868    ROUND(12.9937, 3) ==&gt; 12.994          ROUND(13252.0, -2) ==&gt; 13200.0    ROUND(1432.99, -2) ==&gt; 1400.00    ROUND(12.99,0.7) ==&gt; 13.0          ROUND(56.34, 0.7) ==&gt; 56.3    ROUND(11.99) ==&gt; 12.0    ROUND(-14.99) ==&gt; -15.0</p>		

<b>RowNum</b>	
Description	Returns the current row number of the buffer from the source.
Syntax	RowNum()
Returns	Numeric (Int64)
Examples	
RowNum()	

<b>Rpad</b>		
Description	Converts a string to a specified length by adding blanks or characters to the end of the string.	
Syntax	Rpad( 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.
		Must be a positive integer

Length	false	literal. This argument specifies the length you want each string to be.
<b>Rpad</b>		
Second_String	true	Can be any string value. The characters you want to append to the right-side of the 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		
Rpad("SentryOne", 30), Rpad(CompanyNameColumn, 50)		

<b>RTrim</b>		
Description	Removes blanks or characters from the end of a string.	
Syntax	RTRIM( 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.	
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.
		Any string value. Passes the characters you want to remove from the end of the string. You can also enter a text literal. However, you must

Trim_Set	true <b>RTrim</b>	enclose the characters you want to remove from the end of the string within single quotation marks, for example, 'abc'. If you omit the second string, the function removes blanks from the end of the first string. RTRIM is case sensitive.
Examples		
<code>RTrim("SentryOne "), RTrim(CompanyNameColumn), RTrim("SentryOne----", "-")</code>		