

Task Factory Functions List D-E

Last Modified on 01 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.

DateCompare

Description	Compares two dates against each other.	
Syntax	DateCompare(Date1, Date2)	
Returns	Integer. If Date1 is greater than Date2, 1 is returned. If Date1 equals Date2, 0 is returned. If Date1 is less than Date2, -1 is returned.	
Parameters		
Name	Optional	Description
Date1	false	The first date in the comparison.
Date2	false	The second date in the comparison.
Examples		
DateCompare("7/4/2012", "7/5/2012"), DateCompare(varStartDate, varEndDate), DateCompare(OrderDateColumn, StartDateColumn)		

DateDiff

Description	Returns the difference between two dates against each other.	
Syntax	DateDiff(Date1, Date2, Format)	
Returns	Integer. The difference based on datepart between the two days.	
Parameters		
Name	Optional	Description
Date1	false	The first date in the comparison.
Date2	false	The second date in the comparison.

<h2>DateDiff</h2>		<p>The datepart tells the DateDiff what part of the date to compare.</p> <p>Year = "Y", "YY", "YYY", "YYYY" Day = "D", "DD", "DDD", "DY", "DAY" Month = "M", "MM", "MON", "MONTH" Hour = "H", "HH", "HH12", "HH24" Minute = "MI" Second = "S", "SS" Millisecond = "MS" Microsecond = "US" Week Of Year = "W", "WW" Quarter = "Q", "QQ"</p>
Format	false	
Examples		
<pre>DateDiff("7/4/2012", "7/5/2012", "d"), DateDiff(varStartDate, varEndDate, "M"), DateDiff(OrderDateColumn, StartDateColumn, "H")</pre>		

<h2>DatePart</h2>		
Description	Returns the part of the date specified in the datepart parameter.	
Syntax	DatePart(Date, Date_Part)	
Returns	Object.	
Parameters		
Name	Optional	Description
Date	false	The date used to retrieve the datepart from.
Date_Part	false	<p>The datepart tells the DatePart what part of the date to retrieve.</p> <p>Year = "Y", "YY", "YYY", "YYYY" Day = "D", "DD", "DY", "DAY" Month = "M", "MM", "MON", "MONTH" Hour = "H", "HH", "HH12", "HH24" Minute = "MI" Second = "SS" Millisecond = "MS" Microsecond = "US" Day Of Week (1-7) = "WD", "W" Week Of Year = "WW" Day Of Year = "DDD", "YD" Quarter = "Q" Julian Date = "J"</p>
Examples		
<pre>DatePart("7/4/2012", "H") DatePart("7/5/2012", "MM") DatePart("7/5/2012", "MM")</pre>		

```
DatePart("1/4/2012", "d"), DatePart(varStartDate, "MM"), DatePart(OrderDateColumn, "HH")
```

DatePart

Day

Description	Returns the Day from the date specified.	
Syntax	Day(Date)	
Returns	returns integer value of 1 - 31	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve day from
Examples		
Day("12/8/2011"), Day(varDate), Day(OrderDateColumn)		

Day_Of_Week

Description	Returns the Day of the week from the date specified.	
Syntax	Day_Of_Week(date)	
Returns	returns integer value of 1 - 7	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve day of week from
Examples		
Day_Of_Week("12/8/2011"), Day_Of_Week(varDate), Day_Of_Week(OrderDateColumn)		

DayName

Description	Returns the name of the day (Monday, Tuesday, etc) from the date specified.	
Syntax	DayName(Date, Return_Shortname)	
Returns	returns string: Name of the day	

returns	returns string. Name of the day	
DayName		
Parameters		
Name	Optional	Description
Date	false	Date to retrieve day name from
Return_Shortname	false	Return the short name of the day (Mon, Tue, Wed)
Examples		
DayName("12/8/2011", false), DayName(varDate, True), DayName(OrderDateColumn, false)		

DayOfYear		
Description	Returns the day of the year of the date specified.	
Syntax	DayOfYear(Date)	
Returns	returns integer value of 1 - 366	
Parameters		
Name	Optional	Description
Date	false	Date to retrieve day of year from
Examples		
DayOfYear("12/8/2011"), DayOfYear(varDate), DayOfYear(OrderDateColumn)		

DD_DELETE	
Description	DD_DELETE is equivalent to the integer literal 2.

DD_INSERT	
Description	DD_INSERT is equivalent to the integer literal 0.

DD_REJECT

Description	DD_REJECT is equivalent to the integer literal 3.
-------------	---

DD_UPDATE

Description	DD_INSERT is equivalent to the integer literal 1.
-------------	---

DecBase64

Description	Decodes a base 64 encoded value and returns a string with the binary data representation of the data. If you encode data using EncBase64, and you want to decode data using DecBase64, you must run the decoding dataflow using the same data movement mode. Otherwise, the output of the decoded data may differ from the original data.	
Syntax	DecBase64(Value)	
Returns	Binary decoded value. NULL if the input is a null value. Return values differ if you run the dataflow in Unicode mode versus ASCII mode.	
Parameters		
Name	Optional	Description
Value	false	String datatype. Data that you want to decode.
Examples		
example goes here		

DecimalToBin

Description	Converts a decimal value to an binary value.	
Syntax	DecimalToBin(Value)	
Returns	String	
Parameters		

Name	Optional	Description
value	false	The decimal value to convert.
Examples		
DecimalToBin(10)		

DecimalToHex

Description	Converts a decimal value to an hexadecimal value.	
Syntax	DecimalToHex(Value)	
Returns	String	
Parameters		
Name	Optional	Description
Value	false	The decimal value to convert.
Examples		
DecimalToHex(10)		

DecimalToOct

Description	Converts a decimal value to an octal value.	
Syntax	DecimalToHex(Value)	
Returns	Int64	
Parameters		
Name	Optional	Description
Value	false	The decimal value to convert.
Examples		
DecimalToOct(10)		

Decode

Description	Searches a string for a value you specify. If the function finds the value, it returns a result value, which you define. You can build an unlimited number of searches within a DECODE function.	
Syntax	DECODE(Value, First_Search, First_Result [, Second_Search, Second_Result]...[, Default])	
Returns	First_result if the search finds a matching value. Default value if the search does not find a matching value. NULL if you omit the default argument and the search does not find a matching value. Even if multiple conditions are met, the Integration Service returns the first matching result. If the data contains multibyte characters and the DECODE expression compares string data, the return value depends on the code page and data movement mode of the Integration Service.	
Parameters		
Name	Optional	Description
Value	false	Any datatype except Binary. Passes the values you want to search. You can enter any valid task editor.
Search	false	Any value with the same datatype as the value argument. Passes the values for which you want to search. The search value must match the value argument. You cannot search for a portion of a value. Also, the search value is case sensitive. For example, if you want to search for the string 'Halogen Flashlight' in a particular port, you must enter 'Halogen Flashlight, not just 'Halogen'. If you enter 'Halogen', the search does not find a matching value. You can enter any valid task editor.
Result	false	Any datatype except Binary. The value you want to return if the search finds a matching value. You can enter any valid task editor.
Default	true	Any datatype except Binary. The value you want to return if the search does not find a matching value. You can enter any valid task editor.
Examples		
Decode(ProductCode, "R", "Rims", "S", "Silver Lining", "M", "Mats")		

Decode DeleteFilesOlderThanNDays

Description	Deletes from from a directory that are N days old.	
Syntax	DeleteFilesOlderThanNDays(Path, Pattern, Older_Than_Days, Recursive [, Continue_On_Failure])	
Returns	Integer. Returns the number of files deleted	
Parameters		
Name	Optional	Description
Path	false	The path where the file will be deleted from.
Pattern	false	The filter pattern for locating and deleting files. *.* = all files, *.xls= all xls files, 2012*.doc = any file that starts with 2012 and ends with .doc.
Older_Than_Days	false	The number of days that will be used to delete files older than.
Recursive	false	Delete files from subdirectories as well.
Continue_On_Failure	false	Continue deleting files even if one or more files fails to delete.
Examples		
DeleteFilesOlderThanNDays("C:\ssis\test", "*.xml", 14, false, true)		

DirCreate

Description	Creates a directory in the path supplied.	
Syntax	DirCreate(Path)	
Returns	Boolean. True if the Directory was created. False if it was not or path parameter is null	
Parameters		
Name	Optional	Description
Path	false	The path where the directory will be created.

Examples	<h2>DirCreate</h2>
DirCreate("C:\ssis\test")	

<h2>DirDelete</h2>		
Description	Removes a directory in the path supplied.	
Syntax	DirDelete(Path)	
Returns	Boolean. True if the Directory was removed. False if it was not or path parameter is null	
Parameters		
Name	Optional	Description
Path	false	The path where the directory will be removed from.
Examples		
DirDelete("C:\ssis\test")		

<h2>DirExists</h2>		
Description	Check if the directory exists in the path supplied.	
Syntax	DirExists(Path)	
Returns	Boolean. True if the Directory exists. False if it does not or path parameter is null	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Examples		
DirExists("C:\ssis\test")		

<h2>DirGetCreationTime</h2>		

Description	Return the creation time of the directory.	
Syntax	DirGetCreationTime(Path).	
Returns	DateTime. If the path does not exist, the minimum date for your system is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Examples		
DirGetCreationTime("C:\ssis\")		

DirGetLastAccessTime

Description	Return the time the directory was last accessed.	
Syntax	DirGetLastAccessTime(Path)	
Returns	DateTime. If the path does not exist, the minimum date for your system is returned.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Examples		
DirGetLastAccessTime("C:\ssis\")		

DirGetLastWriteTime

Description	Return the time the directory was last written to.	
Syntax	DirGetLastWriteTime(Path)	
Returns	DateTime. If the path does not exist, the minimum date for your system is returned.	
Parameters		
Name	Optional	Description

Path	false	The path where the directory is located.
DirGetLastWriteTime		
Examples		
DirGetLastWriteTime("C:\ssis\")		

DirGetRoot		
Description	Returns the root directory of the path specified.	
Syntax	DirGetRoot(Path)	
Returns	String. Null if the path parameter is Null	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Examples		
DirGetRoot("C:\ssis\test")		

DirMove		
Description	Moves a directory in from one path to another.	
Syntax	DirMove(Path_Source, Path_Destination)	
Returns	Boolean. True if the Directory was moved. False if it was not or either the Path_Source or Path_Destination parameters are null	
Parameters		
Name	Optional	Description
Path_Source	false	The path where the directory will be moved from.
Path_Destination	false	The path where the directory will be moved to.
Examples		
DirMove("C:\ssis\test", "C:\ssis\test1")		

DirSetCreationTime

Description	Set the time the directory was created.	
Syntax	DirSetCreationTime(Path, Date_Created)	
Returns	Boolean. True if the value is set. False if either parameter is null.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Date_Created	false	The date the directory was created.
Examples		
DirSetCreationTime("C:\ssis\", "7/6/2012 14:24")		

DirSetLastAccessTime

Description	Set the time the directory was last accessed.	
Syntax	DirSetLastAccessTime(Path, Date_Last_Access)	
Returns	Boolean. True if the value is set. False if either parameter is null.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Date_Last_Access	false	The date the directory was last accessed.
Examples		
DirSetLastAccessTime("C:\ssis\", "7/6/2012 14:24")		

DirSetLastWriteTime

--	--

Description	Set the time the directory was written to accessed.	
Syntax	DirSetLastWriteTime (Path, Date_Last_Written)	
Returns	Boolean. True if the value is set. False if either parameter is null.	
Parameters		
Name	Optional	Description
Path	false	The path where the directory is located.
Date_Last_Written	false	The date the directory was last written to.
Examples		
DirSetLastWriteTime("C:\ssis\","7/6/2012 14:24")		

EncBase64

Description	Encodes data by converting binary data to string data using Multipurpose Internet Mail Extensions (MIME) encoding. Encode data when you want to store data in a database or file that does not allow binary data.	
Syntax	EncBase64(Value)	
Returns	Encoded value. NULL if the input is a null value.	
Parameters		
Name	Optional	Description
Value	false	Binary or String datatype. Data that you want to encode.
Examples		
example goes here		

EndsWith

Description	EndsWith determines whether a string ends with a character or string value	
Syntax	EndsWith (String_To_Search, Search_Value)	
Returns	Boolean. True if string_to_search ends with search_value. NULL if a	

value passed to the function is NULL.		
Parameters		
EndsWith		
Name	Optional	Description
String_To_Search	false	Character string. The string to search.
Search_Value	false	Character string. The value to find at the end of String_To_Search
Examples		
EndsWith("400 College Dr", "Dr"), If EndsWith(AddressColumn, "Dr") Then ...)		

Error		
Description	Causes the Integration Service to skip a row and issue an error message, which you define. The error message displays in the progress log.	
Syntax	ERROR(String)	
Returns	String.	
Parameters		
Name	Optional	Description
String	false	String value. The message you want to display when the Integration Service skips a row based on the expression containing the ERROR function. The string can be any length.
Examples		
Error("Whoops! Something went wrong!")		

ExecuteSQL	
Description	Executes supplied sql statement or stored procedure and returns single value or full resultset depending on column_name_or_index setting. By default it returns first column of first row in the resultset. If you execute DDL (e.g. CREATE TABLE...) or DML (UPDATE/DELETE...) statements which doesnt return anything then it will return NULL. You can use this function to call simple or parameterized SELECT queries or stored procs. You can also call DDL/DML statemets (e.g. UPDATE/DELETE/ALTER/INSERT/TRUNCATE)

Syntax	ExecuteSQL(connection_manager, sql_statement [, is_stored_procedure] [, command_timeout] [, column_name_or_index [, param1, param2 ... paramN])	
Returns	Returns single value or full resultset as ADO.net DataTable. If no result set or value found (e.g. UPDATE/DELETE Statement) then returns NULL	
Parameters		
Name	Optional	Description
Connection_Manager	false	The Connection Manager which will use to execute provided sql_statement. It has to be either OLEDB or ADO.net connection manager. Syntax for connection manager is @@[connectionname]. When you use OLEDB connection to execute queries then you cant define named parameters (e.g. @mypara). You have to use "?" for parameter (Example: select * from customer where customerid=?). When you use ADO.net connection type then you can use named parameters in your queries (e.g. select * from customer where customerid=@in_customer)
Sql_Statement	false	String value used representing the Stored Procedure that will be executed.
Is_Stored_Procedure	true	Boolean value indicating whether the Sql_Statement is a Stored Procedure or not. When this parameter is True then you dont have to include parameters as part of sql_statement. (Default=False)
Command_Timeout	true	Timeout in seconds for sql command. 0 means unlimited (Default=0)
Column_Name_OR_Index	true	If you have more than one column in output rowset and you want to return other than first column then specify column by name or Index (starting from 0). If you pass "- 1" (in double quotes) then it will return full result set as ADO.net DataTable which you can store in object datatype variable and consume it later in script task or use it in ForEach Loop
		List of parameters for supplied

<p>Parameters</p>	<h2 style="text-align: center;">ExecuteSQL</h2> <p style="text-align: center;">true</p>	<p>sql statement. If sqlstatement/stored proc doesn't require params then omit this argument. Parameters are passed using below functions.</p> <p>InParam(paraName [,DataType] [,Precision/Length] [,Scale] , Value) : This function defines Input parameter</p> <p>OutParam(paraName ,DataType ,Precision/Length , Scale , variableNameOrVariablePortName) : This function defines Output parameter</p> <p>Examples: ----- ----- OLEDB CONNECTION EXAMPLE ----- Single Input and output Parameter for OLEDB connection (Use ? for parameter indicator. Named parameters not supported with OLEDB). Returns value back to User Variable "var2" ExecuteSQL(@@[NorthwindOLEDB], "SELECT ?=COUNT(*) FROM ORDERS WHERE CUSTOMERID=?" ,FALSE ,OutParam("out1",DbType.Int32,0,0,"User -- See variable name is in Double quotes.. don't pass as @[var2] ,InParam("prod1","ALFKI")) ---- ----- ADO.NET CONNECTION EXAMPLE ----- ----- Single Input and output Parameter for ADO.net connection (Use @ prefix for named parameters). Returns value back to Variable Column V_COUNT (only for DataFlow transforms) ExecuteSQL(@@[NorthwindOLEDB], "SELECT @MyCount=COUNT(*) FROM ORDERS WHERE CUSTOMERID=@pCustID " ,FALSE ,OutParam("MyCount",DbType.Int32,0,0, V_COUNT) ---See column name is not in DOUBLE QUOTES. This column must be variable column (Not output column) ,InParam("pCustID","ALFKI")) Here is list of supported datatypes which can be used with InParam or OutParam functions http://msdn.microsoft.com/en-us/library/system.data.dbtype.aspx AnsiString = 0, Binary = 1, Byte = 2, Boolean = 3, Currency = 4, Date = 5, DateTime = 6, Decimal = 7, Double = 8, Guid = 9, Int16 = 10, Int32 = 11, Int64 = 12, Object = 13, SByte = 14,</p>
-------------------	---	--

	<h1>ExecuteSQL</h1>	Single = 15, String = 16, Time = 17, UInt16 = 18, UInt32 = 19, UInt64 = 20, VarNumeric = 21, AnsiStringFixedLength = 22, StringFixedLength = 23, Xml = 25, DateTime2 = 26, DateTimeOffset = 27
--	---------------------	--

Examples

```

===== Examples (Calling DML/DDDL Statements e.g.
CREATE/UPDATE/INSERT/DELETE): ===== --Call simple sql
statement which doesn't contain any paramter ExecuteSQL(@@[NorthwindOLEDB] , "DELETE FROM
Customer Where Country='USA'") --Call simple create table ExecuteSQL(@@[NorthwindOLEDB] ,
"CREATE TABLE MyTable( col1 int, col2 int)") ===== Examples
(Returning data from stored proc or select query): ===== --
Return first column of first row ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where
Country='USA'") --Return "Phone" column from 1st row of resultset [Use of Column Name]
ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country='USA' ",False,0,
"Phone") --Call stored proc and return "data" column from first row of sp_spaceused output
ExecuteSQL(@@[NorthwindOLEDB] , "exec sp_spaceused 'dbo.Customers' ",False,0, "data") --Return 5th
column from 1st row of resultset [Use of ColumnIndex] ExecuteSQL(@@[NorthwindOLEDB] , "SELECT *
FROM Customer Where Country='USA' ",False,0, 5) --Return Full Datatable (Call from ExpressionTask.
Bind Expression with variable with object datatype to store full DataTable)
ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country='USA' ",False,0, "-1") --
Timeout query if it takes more than 30 seconds ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM
Customer Where Country='USA' ",False,30) ===== Parameter
Examples (OLEDB Connection): ===== --Single Input
Parameter for OLEDB connection ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer
Where Country=? " ,False, InParam("para1",DbType.String,40,0, "USA" ) ) --Single Input Parameter for
OLEDB connection (Don't pass datatype info.. and let it guess)... If it can't guess it woll use String(4000)
ExecuteSQL(@@[NorthwindOLEDB] , "SELECT * FROM Customer Where Country=? " ,False,
InParam("para1", "USA" ) ) --Single Input and output Parameter for OLEDB connection. Returns value back
to User Variable "var2" ExecuteSQL(@@[NorthwindOLEDB] , "SELECT ?=COUNT(*) FROM ORDERS WHERE
CUSTOMERID=?" ,FALSE ,OutParam("out1",DbType.Int32,0,0,"User::var2") -- See variable name is in
Double quotes.. don't pass as @[var2] ,InParam("prod1","ALFKI") ) --Return value from stored procedure
(RETURN VALUE always numeric and output from stored proc via RETURN statement) --Single output
Parameter for OLEDB connection. Returns value back to User Variable "var2"
ExecuteSQL(@@[NorthwindOLEDB] , "exec ?=sp_MyStoredProc " ,FALSE
,OutParam("out1",DbType.Int32,0,0,"User::var2") -- See variable name is in Double quotes.. don't pass as
@[var2]. ) --Single Input and output Parameter for OLEDB connection. Returns value back to Variable
column V_COUNT ( only for DataFlow transforms) ExecuteSQL(@@[NorthwindOLEDB] , "SELECT
@MyCount=COUNT(*) FROM ORDERS WHERE CUSTOMERID=@pCustID " ,FALSE
,OutParam("MyCount",DbType.Int32,0,0, V_COUNT) ---See port name is not in DOUBLE QUOTES
,InParam("pCustID","ALFKI") ) ===== Parameter Examples
(ADO.net Connection): ===== --Single Input Parameter for
ADONET connection ExecuteSQL(@@[NorthwindADONET] , "SELECT * FROM Customer Where
Country=@pCountry" ,False, InParam("pCountry",DbType.String,40,0, "USA" ) ) --Single Input and output
Parameter for ADONET connection. Returns value back to User Variable "var2"
ExecuteSQL(@@[NorthwindADONET] , "SELECT @MyCount=COUNT(*) FROM ORDERS WHERE
CUSTOMERID=@pCustID " ,FALSE ,OutParam("MyCount",DbType.Int32,0,0,"User::var2")
,InParam("pCustID","ALFKI") ) --Single Input and output Parameter for ADONET connection. Returns value
back to Variable column V_COUNT ( only for DataFlow transforms) ExecuteSQL(@@[NorthwindADONET] ,
"SELECT @MyCount=COUNT(*) FROM ORDERS WHERE CUSTOMERID=@pCustID " ,FALSE
,OutParam("MyCount",DbType.Int32,0,0, V_COUNT) ,InParam("pCustID","ALFKI") )

```

Exp

Description	Returns Euler's Number, e, raised to the specified power (Exponent), where e=2.71828183. For example, EXP(2) returns 7.38905609893065. You might use this function to analyze scientific and technical data rather than business data. EXP is the reciprocal of the LN function, which returns the natural logarithm
-------------	--

	of a numeric value.	
Syntax	EXP(Exponent)	
Returns	Double value. NULL if a value passed as an argument to the function is NULL.	
Parameters		
Name	Optional	Description
Exponent	false	Numeric datatype. The value to which you want to raise e. The exponent in the equation e^{value} . You can enter any valid task editor.
Examples		
EXP(ExponentColumn)		