Standard Functions

Standard Functions, or built-in functions, refer to specific functions by name. Functions are available to all users of a project. If a project is imported, the same files and functions, from the original project, is available to the cloned project.

> **Note**
> 1) Functions are code sensitive.
> 2) When you use functions, the string objects must be enclosed in single quotes.

Function	Description	Examples
ADDXML(String xml)	This function adds XML structure to the present element. This function is used when the source application doesn't expose their fields in their API. A user can map the field by modifying the XML structure to post back. This will add the XML on the root node of the target XML structure.	Mapping: <TargetField> = { ADDXML("< "</xml element>");return "";} **Example: Database fields** **Trigger:** select id,firstname from Contac The above query fetches id, firstname fr The table also has "contactaddress" fiel "contactaddress" field in the target sche **Name = {ADDXML("<address>" +VAL** However, the source field can be mappe into contactaddress column of the datab
ADDXML(String xPath, String xml)	This function appends the XML structure to the XPath passed. This function is used when the source application doesn't expose fields through API. A user can map the field and parse the data.	Let us take an example of custom field S Map this field to the SalesRep custom fi To map this field, type custom field nam (QuickBook online). And then, add the fo **Mapping: <TargetField> = {** **ADDXML("<CustomField><Definitionl** **return "";** **}** Where**:** **<DefinitionId>2</DefinitionId>** :Positio **<Name>SalesRep</Name>** : The custo **<StringValue>"+VALUE("SalesRep_**
AND (boolean booleanExp1, boolean booleanExp2)	This function is used to check more than one condition/expression at the same time. It returns true only if both the conditions are met; or else, it returns false.	Mapping: <TargetField> = AND(boolean **Example : <TargetField> = AND(VALU** **VALUE("Type")=="Key Account")** From the above example, the function re the "Stage" is "Closed Won" and, the Ty

ASNUMBER(String value)	This function checks if a supplied value is a number. And then, it returns the actual number passed; or else, it returns zero.	MAPPING: <TargetField> = ASNUMBEI **<TargetField> = ASNUMBER("123")** If the value for the number of employees a value zero. In this example, the value the function will return 123.
CLEAN(String s1, String s2)	This function removes all non-printable characters from a supplied string value. **Note:** The clean function removes the first 32 (non-printable) characters in the 7-bit ASCII code from the text.	Mapping: <TargetField> = CLEAN("Strin **Example: <TargetField> = CLEAN (A°** In the above example, the function remo and returns the output as "Alan".
CONCATENATE(String... strings)	This function allows you to join two or more text strings together.	Mapping: <TargetField> = CONCATENA **Example: passing text values** **<TargetField> = CONCATENATE("It's** The value returned by the function from
CODE(String s)	This function returns the numeric code of the first character of a supplied text value.	Mapping: <TargetField> = CODE("text v **Example : <TargetField> = CODE("Ala** In the above example the function returr in the supplied text - namely, "Alan Musl "A" is returned as 65.
DATE(String dateString)	This function reads a date string and returns it in the format of "yyyyMMdd-HHmmssZ".	Mapping: <TargetField> = DATE("date s **Example : <TargetField> = DATE("03-** The value returned would be transforme
DATE (String dateString, String inputFormat)	This function reads a date string and returns the date string as per user-specified date format. If the user-specified date format is not passed then, it passes date string as "yyyyMMdd-HHmmssZ".	Mapping: <TargetField> = DATE(VALUE **Example** : **<TargetField> = DATE(VAL** The value of lasttransferdate is Mar-12-2 the **DATE()** function will return value as
DATE(String dateString, String inputFormat, String outputFormat)	This function reads a date string and returns the date string as per user-specified date output format. If the user-specified date input format is not passed then it returns date string as "yyyyMMdd-HHmmssZ".	Mapping: <TargetField> = DATE(VALUE **Example : <TargetField>= DATE(DATI** **"yyyy-MM-dd'T'HH:mm:ss'Z'")** The value of createddate is 03-12-2010 function will return value as 2010-12-03'
DOLLAR(Arg 0)	This function is currently not supported.	

EQUALS(String v1, String v2)	This function compares two string values and returns true if both the values are an exact match; otherwise, it returns false(case-insensitive).	Mapping: <TargetField> = EQUALS(VAI **Example: <TargetField> = EQUALS("s** From the above example the function re and the second string does not match.
ERROR()	This function can be used to get the error message for errors during writing to target. This is used in status write back, once an attempt has been made to write the record to the target.	Mapping: <TargetField> = ERROR() Example: There is an invalid reference t "Dialysis Center of NW Arkansas: Hidde SalesOrder. QuickBooks error message The specified record does not exist in th From the above example, there is an err QuickBooks sales order and this error m
EXACT(String str1, String str2)	This function compares two string values and returns true if both the values are an exact match; otherwise, it returns false(case-sensitive).	Mapping: <TargetField> = EXACT(VALU **Example : <TargetField> = EXACT(VA** The above example will return true as th value of second attribute is also "Chris".
FAST_TLOOKUP(String query)	This function looks up a given Id from a passed query and returns the corresponding value. This function is the same as the TLOOKUP function. However, the function executes based on writer batch size for faster execution. Note: You shouldn't use the special character on the function.The return value of FAST_TLOOKUP 1. should not start with a hyphen 2. should not end with hyphen and 3. should not be Null.	Mapping: <TargetField> = FAST_TLOO **Example: QuickBooks Invoice to Sale** **AccountID = FAST_TLOOKUP("Selec** In the above example, we are updating t by reading invoice records from QuickBo will fetch the customer List ID (in this cas If a match is found between List id and S that the customer's record exists in Sale
FIND(String findText, String withinText)	This function returns the position of the first string parameter, within the supplied second string parameter.	Mapping: <TargetField> = Find(VALUE(**<TargetField> = FIND("arch","search"** In this example, pass the first string as " The **FIND()** function will return the positi The FIND() function will return 3 as the v **Note:** If the find text parameter finds val the function returns the position of the ve

FIND(String findText, String withinText, int startNum)	This function returns the position of the first string parameter within the supplied second string parameter from the passed start index.	Mapping: TargetField = FIND(VALUE("S **TargetField = FIND("arch","search",2)** In this example, pass the first string as " The **FIND()** function will search the strin the 2nd position of the "search" string ar The function return 3 and first two chara **Note:** If the findText parameter finds val then the function returns the position of t
FIXED (Double)	This function rounds the decimal values up to two digits and returns the round off value.	Mapping: <TargetField> = FIXED("sourc **<TargetField> = FIXED(12.6789)** In the price field the value passed is 12. will return the price value as 12.68.
FIXED (Double, Integer)	This function rounds a supplied number to a specified number of decimal places.	Mapping: <TargetField> = FIXED("sourc **<TargetField> = FIXED(14.789,1)** In the price field the value passed is 14.
FIXED (Double, Integer, Boolean)	This function is currently not supported	
FORMAT(String value, String format)	This function transforms the numeric string passed in the first parameter based on the format passed in the second parameter. Then, it returns the transformed number as a string.	Mapping: <TargetField> = FORMAT("Sc **<TargetField> = FORMAT("12","0.00#** In the above example, pass the first para
GETROOTVALUE(String elementName)	This function returns the immediate parent node of XML element.	Mapping: <TargetField> = GETROOTVA **Example : <TargetField> = GETROOT** <Contact> <Name>John</Name> </Contact> The "Name" attribute will be compared i
GETSOURCEXML2STRING ()	This function returns the string formatted XML structure of the row.	Mapping: <TargetField> = GETSOURCE **<TargetField> = GETSOURCEXML2ST** <items> <item id="0001" type="donut"> </item> ... </items> The XML elements are returned as a str

GETSOURCEXML2STRING (String elementName)	This function returns the string formatted XML structure of the row, for the element name passed.	Mapping: <TargetField> = GETSOURCE **Example : <TargetField> = GETSOUR** <items> <item id="0001" type="donut"> id="5001">None</topping> <topping id= The "batters" node element is found in th <batters> <batter id="1001">Regular</b
IF(Boolean condition, String trueValue, String falseValue)	This function tests the user-defined condition and returns one result if the condition is true, and another if the condition is false.	Mapping : <TargetField> = IF(String,Inte **<TargetField> = IF(ISEMPTY(VALUE("** In this case, the **IF** function checks for th
ISEMPTY(String s)	This function returns true if the variable is unintialized or explicity set to empty; otherwise, it returns false.	Mapping: <TargetField> = ISEMPTY("sc **<TargetField> = ISEMPTY(VALUE("Fi** In the above example, the first name fiel
ISNULL(String s)	This function checks if the passed value is null then, it returns true; or else, it returns false.	Mapping: <TargetField> = ISEMPTY("sc **<TargetField> = ISNULL(VALUE("last** In the above example, the first name fiel
LEFT (String var)	This function returns the first character of the supplied string which is on left hand side.	Mapping: <TargetField> = LEFT(VALUE **Example: passing a string** **<TargetField> = LEFT("search")** In this example, pass a string as "search
LEFT(String var, int count)	This function returns the specified number of characters from the left of the given string.	Mapping: <TargetField> = LEFT(VALUE **Example: passing a string** **<TargetField> = LEFT("search",3)** In this example, pass the first parameter "sea". **Note:** If the length of the passed string is
LEN(String var)	This function returns the length of a given string.	Mapping: <TargetField> = LEN(String) **Example : passing a string** **<TargetField> = LEN("search")** In this example, pass the string as "sear

LINK(String colName, String value) **Note:** This function is deprecated and an alternative function could be MEMLOOKUP orTLOOKUP functions.	This function reads the second parameter passed through this function and sets it as an attribute to the column name which is passed through the first parameter.	Mapping: <TargetField> = LINK(VALUE(**<TargetField> = LINK(VALUE("Name"** In the "Name" node, the "Id" value will b

LINK(String colName, String query, String fieldName, String refValue, String objectName)
 	
Note: This function is deprecated and an alternative function could be MEMLOOKUP orTLOOKUP functions.
	The Link function is used to update Salesforce (Target) field with a source value based on the source identifier that remains unique across **Salesforce** (Target) as well as **Source** (Ex: Quickbooks) application.
	
	The function creates an internally cached table querying the target (Salesforce) database with two columns - the first column being the primary key for the table; and, the second for the other column name.
	
	It then looks for a string that is the same across both the Source and Target system before updating the Target LOOK-UP(Salesforce) field with the Source field value.
		Example: Mapping:
		
		`AVSFQuickBooks__Op`
		`"Select id,AVSFQui`
		`"AVSFQuickBooks__Q`
		`"Opportunity")`
		
		`In the above examp`
		`look-up field AVSF`
		`where AVSFQuickBoo`
		`transaction Id "Tx`
		`Salesforce as well`
		
		`In order to update`
		`we are creating a`
		`Then it does a loo`
		`Salesforce field A`
		`with that of Quick`

LINK(String colName, String query, String fieldName, String refValue, String objectName, String valueField) **Note:** This function is deprecated and an alternative function could be MEMLOOKUP orTLOOKUP functions.	The function creates an internally cached table querying the target (Salesforce) database with two columns - first column being the primary key for the table; and, the second for the other column name.	Mapping: <TargetField> = LINK(VALUE(**Example : <TargetField> = LINK(VALU** In "Name" node, default attribute values here it is "Id", will be returned back.
LOG()	This function is used to print the value of a parameter passed to the DBSync console. It can be used for debugging.	Mapping: <TargetField> = LOG(VALUE(**Example : <TargetField> = {LOG("**** The log function gets the value of the ta

LOOP	This function is different from the conventional '=' operator to map source and target. <SourceField **"loop"** TargetField> is used in scenarios where grouping the Line Items are required based on the target identifier. It can also be used when the identifier is unique and the same across both source and target systems.	**Example1: Salesforce Opportunity Li** Mapping <TargetField> : InvoiceAddRq/ In the above example, for every opportu **Example2: Source - Database ; Targe** Database Query : select invoice_no,cus Mapping : InvoiceAddRq/InvoiceAdd/Inv In the above example, we are using the Quickbooks Invoice Line Item. For instance, if the database Invoice tab rather than, creating one Invoice for eac
LOOKUP(String adapterName, String queryString)	This function searches for a specific value in the source connector where the condition is passed in the form of a query to be searched on the first parameter.	Mapping: <TargetField> = LOOKUP("so **Example:<TargetField> = LOOKUP("S** The above example with fetch the Id fro
LOWER (String var)	This function converts all characters in a given string to lowercase.	Mapping: <TargetField> = LOWER(Strin **Example: passing a string** In this example, pass a string as "SEAR **<TargetField> = LOWER("SEARCH")** The function will return "SEARCH" as "s
LPAD(String text, String pattern, int pad)	This function is used to pad the left side of a string with a specific set of characters. The integer is the total length of the string returned after padding.	Mapping: <TargetField> = LPAD(VALUE **Example1: <TargetField> = LPAD("tec** The function will return result as "tech or **Example2: <TargetField> = LPAD("tec** The function will return the result as "00(
LSPLIT(String text, String splitter)	This function returns the split value of the string starting from the left side of a string till the splitter value.	Mapping: <TargetField> = LSPLIT(VALU **Example: <TargetField> = LSPLIT ("te** The **LSPLIT()** would return the result as
MAP(String key, String...mapEntries)	This function returns the value against the key passed from the key-value pair(s) passed via the second parameter of the function.	Mapping: <TargetField> = MAP(VALUE(**Example: <TargetField> = MAP("myK** The MAP function will return "myValue".

MEMTABLE(String cacheIdentifier, String query)	This function creates a data cache in the system. This data cache will be referred to using cacheIdentifier. Once MEMTABLE function is called, the cacheIdentifier passed can be used to search data in the query (executed against the target connector).	MAPPING: MEMTABLE(StringCacheIde **Example: Database Table** **MEMTABLE("AccountID","Select ID, I** In this example, a cache with the name ' Assuming we have a_01Name1, b_01N
MEMLOOKUP(String cacheIdentifier, String key)	This function returns value against the key in the dataset referred by the cacheIdentifier.	Mapping: <TargetField> = MEMLOOKU **Example:** **<TargetField> = MEMLOOKUP("Acco** In this example, AccountID is the name This function call will return "Name1" as cacheIdentifier. In ideal scenarios, second parameter wi For MEMLOOKUP to work, we need to
MEMLOOKUPREGEX(String cacheIdentifier, String key, String regex)	This function is similar to MEMLOOKUP function with an additional check of a regular expression against the key name. If the passed key matches the pattern passed in the regex, it will return the corresponding value; otherwise, it will return an empty string.	Mapping: <TargetField> = MEMLOOKU **Example:** **<TargetField> = MEMLOOKUP("Acco** The above adds an additional regex par doesn't start with "a_".
MID(String text, Int startNum, Int numChars)	This function extracts a substring from the string and returns the substring.	Mapping: <TargetField> = MID(VALUE(' **Example : <TargetField> = MID(VALU** In this example, the value of "Firstname'
NOTEQUALS(String v1, String v2)	This function compares the value with another value and returns true if it is not equal; or else, it returns false.	Mapping: <TargetField> = NOTEQUALS **Example : <TargetField> = NOTEQUA** This function compares "USD" against a
OR(boolean exp1, boolean exp2)	This function evaluates the conditions passed through the function and returns true if any one of the condition evaluates to true; otherwise, it returns false.	Mapping: <TargetField> = OR(Boolean, **Example : <TargetField> = OR(VALUE** In the above function, the 'firstname' has returns false.
PARAM(String name)	This function PARAM extracts the values from the session which is in the format PARAM.SOURCE_Object.Variable=PARAM.TARGET_Object.Variable and returns the variable value.	Mapping: <TargetField> = PARAM("strin **Example :<TargetField> = PARAM(VA** If the value of "Description" starts with "F

PARAM_PARENT(String name)	This function PARAM extracts the values from the session which is in the format PARAM.SOURCE_Object.Variable=PARAM.TARGET_Object/Variable and returns the parent value.	Mapping: <TargetField> = PARAM_PAR **Example: <TargetField> = PARAM_PA** If the value of "Description" starts with "F
PARENTVALUE(String name)	This function reads any node elements and returns the immediate parent value of the node element passed.	Mapping: <TargetField> = PARENTVAL **Example : Salesforce object fields** **Trigger:** select Id, name, account.name The above query will retrieve ID, Name a **<TargetField> = PARENTVALUE("Acc** This function retrieves the value of the A
PARENTVALUEATTR(String path, String attr)	This function reads the passed node element and returns attributes of the immediate parent node.	Mapping: <TargetField> = PARENTVAL **Example : <TargetField> = PARENTV** <CustomerRef> <Name>Alan</Name> </CustomerRef> In this example the CustomerRef object
PROPER(String text)	This function reads a string and converts the first letter of a word to upper case and rest of the alphabets in a word to lower case. This is used to represent camel notation.	Mapping: <TargetField> = PROPER("Sc **Example: <TargetField> = PROPER("s** In this example, since Pass string has "s
REPLACE (String oldText, Int startNum, Int numChars, String newText)	This function replaces a full string, or a part of string text, with another text string from the position sent through parameter i.e. startNum.	Mapping: <TargetField> = REPLACE(VA **Example: Pass a string** **<TargetField> = REPLACE("search",3** In this example, Pass string has - (1) "se "a" as the replacement string. The **REPL** "search".
REPT(String text, Int numberOfTimes)	This function returns a string consisting of a supplied text string, repeated specified number of times.	Mapping: <TargetField> = REPT(VALUE **Example:<TargetField> = REPT("tech** In the above example, the function REP
RIGHT(String var)	This function returns a rightmost character of the string value passed.	Mapping: <TargetField> = RIGHT(String **Example: String Parameter** In this example, the Pass string is "searc **<TargetField> = RIGHT("search",1)**

RIGHT(String var, Int count)	This function returns a specified number of characters from the end of a supplied text string.	Mapping: <TargetField> = RIGHT(VALU **Example: String parameter** In this example, Pass first string is "sear function will be "earch". **<TargetField> = RIGHT("search",5)** **Note:** If a number of characters in the st
RPAD(String var, String value, Int size)	This function returns a string after padding the input string with extra characters from the right side. The user can pass the size of the input string until where the padding should be done.	Mapping: <TargetField> = RPAD(VALUE **Example 1: <TargetField> = RPAD("te** The function will return result string as "t **Example 2: <TargetField> = RPAD("te** The function will return result as "tech or
RSPLIT(String var, String splitter)	This function takes splitter text and compares it with the variable text. The function splits variable text based on the splitter text and returns the number of characters after the splitter text to the right.	Mapping: TargetField = RSPLIT(VALUE **Example: <TargetField> = RSPLIT ("te** This function will return result string as r
SEARCH(String findText, String withinText)	This function returns the position of a supplied text string from within a supplied text string.	Mapping: <TargetField> = SEARCH(VA **Example: String parameters** In this example, Pass the first string is "a **<TargetField> = SEARCH("arch","sea** **Note:** If the search string exists more tha
SEARCH(String findText, String withinText, int startNum)	This function returns the position of a supplied text string from within a supplied text string for which starting position can be specified.	Mapping: <TargetField> = SEARCH(VA **Example : String parameters** **<TargetField> = SEARCH("arch","sea** In this example, Pass the first string is "a the place value of "arch" in "search". So **Note:** If the search string exists more tha
SESSION_GET(String name)	This function returns the value of the key stored in the session of that particular process or workflow.	Mapping: <TargetField> = SESSION_GE **Example: Retrieving the "key" stored** **CustomerAddRq/CustomerAdd/Name** In the above example, we are retrieving Avankia**"**.

SESSION_PUT(String name, String value)	This function stores the key/value pair in the session of an active process or workflow. This function will only work with active workflows within DBSync.	Mapping: <TargetField> = SESSION_PU **Example: Storing a static value (Acco** **out = SESSION_PUT("Account_Name** The example stores a static key/value pa to the console.
SETATTR(String colName, String attName, String attValue, String colVal)	This function sets the column with an attribute of name and value as specified. The column value would be set as specified in the colVal. This function can only be applied when writing to Salesforce for the Pricebook object.	Mapping: <TargetField> = SETATTR(Ta **Example :SETATTR("PricebookEntryI** This function is used to query Pricebook value of pricebookentry id and assigns it
SUBSTITUTE(String str, String oldStr, String newStr)	This function replaces all occurrences of a string, within an old string, with the passed new string.	Mapping: <TargetField> = SUBSTITUTE **Example: <TargetField> = SUBSTITUT** In the above example all of "firstname" fi
SUBSTITUTE(String str, String oldStr, String newStr, int occurrences)	This function replaces the specified number of occurrences of a string, within an old string, with the passed new string.	Mapping: <TargetField> = SUBSTITUTE **Example : TargetField = SUBSTITUTE** In the above example, the firstname field
TEXT(Arg 0, Arg1)	This function is not currently supported.	
TLOOKUP(String queryString)	Returns the value for the column in the query. In the query, only one column can be specified.	Mapping: <TargetField> = TLOOKUP(st **Example: QuickBooks Invoice to Sale** **AccountID = TLOOKUP("Select Id fro** In the above example, we are updating t this case, the value of List id is "12345") Salesforce. As a result, related invoice r
TODAY()	This function returns Today's date.Format returned from TODAY() is "Day Mon DD HH:MM:SS TTT YYYY"	Mapping: <TargetField> = TODAY() **Example**: Fri May 06 07:10:58 CDT 201
TRIM(String value)	This function returns a text value with the leading and trailing, spaces removed.	Mapping: <TargetField> = TRIM(VALUE **Example : <TargetField> = TRIM(VALI** In the above example the field of "firstna
UNIQUEFIELD(String colName, String value, String colValue)	This function sets the column with the column value and attributes with the unique field value and returns as an object.	Mapping: <TargetField> = UNIQUEFIEL **Exmaple : <TargetField> = UNIQUEFII** In "Name" node, the "Id" value will be ac
UPPER(String text)	This function converts all the characters in a passed string to the upper case.	Mapping: <TargetField> = UPPER(String **Example: <TargetField> = UPPER("se** In this example, Pass string is "search".

VALIDATEROW()	This function returns true and is used only for validation of a rule section to check whether any condition is satisfied.	Mapping: <TargetField> = Conditional S **Example: <VALIDATEROW> = IF(ISEM** This condition will execute a row only if t
VALUE(String xPath)	This function converts the input parameter to be read as string and returns the passed value as string.	MAPPING: TargetField = VALUE("Sourc **Example: <TargetField> = VALUE("Na** In the above example the "Name" field is
VALUE(String xPath, boolean treatAsEmpty)	This function converts the input parameter to be read as string and returns the passed values as string. The function returns empty string if the parameter has no value in it.	MAPPING: TargetField = VALUE("Sourc **XML Sample :** <person gender = "female"> <firstname>Alan</firstname> </person> **<TargetField> = VALUE("person/firstr** In the above example, as the "firstname' function returns the string as empty strin
VALUEATTR(String xPath, String attr)	This function takes a path and attribute name. And, it returns the actual value of the attribute.	Mapping: TargetField = VALUEATTR("x **XML Sample :** <person gender = "female"> <firstname>Alan</firstname> </person> **<TargetField> = VALUEATTR("person** In the above example, the function retur