Solution: Understanding Mapping Manager Functions

Purpose and Goal: This document will walk you through TrueCommerce's available functions to use within Mapping Manager, along with examples for each.

Meta Tags: Functions, Business-System Plug-Ins (BSPs), Mapping Manager

Last Revision: 07/27/2022

BSP Functions & examples

1. @SUM(value):string
    a. Function will add up a value.
    b. Example: @SUM([ITEM_QUANTITY])
        i. If two line items were sent on the PO of 4 and 6. The above statement would return the value of 10.
    c. Example: @NUM(@SUM(@NUM([ITEM_QUANTITY]*[ITEM_PRICE])))
        i. Returns the sum of all lines L1(Qty*Price)+L2(Qty*Price)+L3(Qty*Price), etc.)

2. @NUM(real):string
    a. Must assign a token as a number in order to perform a calculation.
    b. Example: @NUM([T:ITEM_UNIT_PRICE]) or @NUM([ITEM_QUANTITY]*[ITEM_UNIT_PRICE])

3. @INT(real,shift, size)
    a. Converts value to an integer, placing "0" for shift and/or size if not needed.
    b. Example: @INT([T:ITEM_QUANTITY],0,5)
        i. If item quantity is "1" returned value would be 00001.

4. @STRING(string):real
    a. All values and calculations will be treated as strings.
    b. Example: @INSTR([R:2.20 PAW_Invoice.Note],TRACKING #)
        i. Finds the first "TRACKING #" in the data and returns a character count up to that point
        ii. If [R:2.20 PAW_Invoice.Note] is "ABC123 TRACKING #", the value returned would be 17.

5. @TRIM LEFT(string):string
    a. Removes all spaces to the left of a field.
    b. Example: @TRIM_LEFT([R:3.4 Item.VendorNum])
        i. [R:3.4 Item.VendorNum]equals " ABCD", returns "ABCD".

6. @TRIM RIGHT(string):string
    a. Removes all spaces to the right of a field.
    b. Example: @TRIM_RIGHT([R:3.4 Item.VendorNum])

        i. [R:3.4 Item.VendorNum] equals “ABCD ”, returns “ABCD”.

7. @TRIM(string):string
    a. Removes all spaces in a field.
    b. Example: @TRIM([R:3.4 Item.VendorNum])
        i. [R:3.4 Item.VendorNum] equals “ AB CD ”, returns “ABCD”.

8. @SWITCH(string1, string2, string 3, …):string
    a. Function will look to the first token set in the statement. If that is not present on the transaction then it will look to the next token set in the statement.
    b. Example: @SWITCH([T:STORE_NUMBER],[T:SHIP_TO_ADDRESS_CODE])
        i. First it will return the store number value. If it is not there then it will return the ship to code.
        ii. Repeats for up to 5 fields until a field contains data.

9. @COPY(string, startpos[int], count [int]):string
    a. You can return specific characters of a token using this function. Also, very useful in basing logic specifically on a value in a token.
    b. Example: @COPY([T:STORE_NUMBER],9,4)
        i. 9 is the starting position value for the store number, and 4 will be the returning value of the statement.
        ii. Store number = 0078742030982 – Returning value of statement above would be 0982.

10. @DATEADD(Date, day[int]):date
    a. Function will add days to a date assigned by the token.
    b. Example: DATEADD([T:DATE_REQUESTED_DELIVERY],2)
        i. Above statement will add two days the token value. If negative two is used it will subtract two days.

11. @SIGN(value):value
    a. Returns a sign (+ or -) depending on the value
    b. Example: @SIGN([T:ITEM_QUANTITY])
        i. @SIGN(100) returns value of +.

12. @INSTR(string, substring)
    a. Finds the starting position of a substring.
    b. Example: @INSTR([R:4.10 Item.UserField1],Tracking #)
        i. This will return the position count where “Tracking #” first appears in the field.
            1. Returns position count of the “T” in “Tracking #”.

13. @LEN(string)
    a. Returns length of field
    b. Example: @LEN([T:ITEM_QUANTITY])
        i. @LEN(100) returns value of 3.

14. @ABS(value):value
    a. This function will give the absolute value of a token.
    b. Example: @ABS([T:ITEM_QUANTITY])
        i. @ABS(-100) would return 100.

15. @STRIP CHARS(String, Chars to Strip):String
    a. Function will strip all characters found in a token.
    b. Example: @STRIP_CHARS([T:SHIP_TO_NAME],#A)
        i. Statement will strip out all “#” and “A” characters found in the ship to name.
        ii. The ‘ character needs to be used to remove commas.

16. @UPPER CASE(String):String
    a. Function will return the data held within a token as all upper case.
    b. Example: @UPPER CASE([T:SHIP_TO_NAME])

17. @LOWER CASE(String):String
    a. Function will return the data held within a token as all lower case.
    b. Example: @LOWER_CASE([T:SHIP_TO_NAME])

18. @REPLACE STRING(String, Search String, Replace String):String
    a. Finds a substring and replaces it.
    b. Example: @REPLACE_STRING([R:1.9 Header.Bill of Lading],Z,A)
        i. If BOL is 1Z2008FQ300, returns value of “1A2008FQ300”

19. @UCC128 PARSE (String, Type):String
    a. Parses sections from the UCC128#
        i. Type Constant:
            1. 1 = Prefix (first 3 digits)
            2. 2 = Suffix (String without first 3 digits)
    b. Example: @UCC128_PARSE([R:1.9 Header.Bill of Lading],1)
        i. If BOL is 1Z2008FQ300, returns value of “1Z2”

20. @REAL ROUND(value, decimal places):real
    a. Function will round a value to the decimal place that is set.
    b. Example: @REAL_ROUND([T:ITEM_UNIT_PRICE],2)
        i. Above statement will return two decimal places for a unit price. If the unit price sent was 5.678, the return value would be 5.68.

21. @REAL TRUNCATE(value, decimal places):real
    a. Function will truncate a value to the decimal place that is set.
    b. Example: @REAL_TRUNCATE([T:ITEM_UNIT_PRICE],2)
        i. Above statement will return two decimal places for a unit price. If the unit price sent was 5.678, the return value would be 5.67.

22. @IF THEN ELSE(value1, compare type, value2, value if true, value if false):value
    a. If the statement is true then it will return the value that is set, else it will perform the function set if the first statement is not true.
    b. A comma is used as an else function. If you wanted a statement to return nothing, then after the first comma add another one.
    c. Example:
       @IF_THEN_ELSE([I: Account ID],=,ACE,[T:DISTRIBUTOR_ADDRESS_CODE],[T:STORE_NUMBER])
        i. If the trading partner is ACE then it will return the value of the DC code, else it will return the store number.
        ii. After ACE below you would enter ACE,, that would return nothing for ACE.

23. @IF OR THEN ELSE(value1, compare type, value2, …, value if true, value if false):value
    a. If either statement is true then it will return the {value if true} that is set, else it will return the {value if false}.
    b. Example:
       @IF_OR_THEN_ELSE([I: Account ID],=,ACE, ([I: Account ID],=, ACE2,[T:DISTRIBUTOR_ADDRESS_CODE],[T:STORE_NUMBER])
        i. If the trading partner is ACE or ACE2 then it will return the value of the DC code, else it will return the store number.

24. @IF AND THEN ELSE(value1, compare type, value2, …, value if true, value if false):value
    a. If both statements are true then it will return the {value if true} that is set, else it will return the {value if false}.
    b. Example:
       @IF_AND_THEN_ELSE([I: AccountID],=,ACE,([T:TRANSACTION_TYPE_CODE],=,SO,[T:DISTRIBUTOR_ADDRESS_CODE],[T:STORE_NUMBER])
        i. If the trading partner is ACE and transaction code type is SO, then it will return the value of the DC code, else it will return the store number.

25. @IS NUMERIC(string)

The INSTR function is one where TrueCommerce staff have typically always needed examples to refer to when writing the statement. Below are the two examples that at least one staff normally uses.

Using the @INSTR(Function to Parse Data

The @INSTR(function provides you with the ability to pull multiple data elements from one field when there is a variable length.

Customer Request 1:

Amazon is sending the first and last name of the customer in one ship to name field. TrueCommerce staff needs to have these separated into a first name and last name field in my system.

Example: “Andrew Cutch”

Solution 1:

We need to answer two questions before writing the statement.

1. What character(s) separate the values?
2. Is this character always present? If the value you are looking for in the @INSTR(is not always present then you will need to include an ‘if’ statement. This ‘if’ statement will check that the character is present.

For this example, the first and last name are separated by a single space and the name is always sent in this format.

First Name – This will copy everything to the left of the space.

@COPY([T:SHIP_TO_NAME],1,@INSTR([T:SHIP_TO_NAME],)-1)

Result: Andrew

Last Name – This will copy everything to the right of the space.

@COPY([T:SHIP_TO_NAME],@INSTR([T:SHIP_TO_NAME],)+1,255)

Result: Cutch

Customer Request 2:

TrueCommerce staff keeps the length, width, and height in 1 field in their system. Their trading partner needs to receive these as three separate values.

Example: 10x11x12

Solution 2:

1. What character(s) separate the values?
2. Is this character always present? If the value you are looking for in the @INSTR(is not always present then you will need to include an 'if' statement. This 'if' statement will check that the character is present.

For this example, the values are separated by an "x" and are always present.

Length - This will copy everything to the left of the first x.

@COPY([R:1.21 Dimension],1,@INSTR([R:1.21 Dimension],x)-1)

Result: 10

Width - This will copy everything between the first and second x.

@COPY(@COPY([R:1.21 Dimension],@INSTR([R:1.21 Dimension],x)+1,255),1,@INSTR(@COPY([R:1.21 Dimension],@INSTR([R:1.21 Dimension],x)+1,255),x)-1)

Result: 11

Height - This will copy everything after the second x.

```
@TRIM(@COPY(@COPY([R:1.21 Dimension],@INSTR([R:1.21 Dimension],x)+1,255)
,@INSTR(@COPY([R:1.21 Dimension],@INSTR([R:1.21
Dimension],x)+1,255),x)+1,255))
```

Result: 12