

Create_Symbolic Routine

Description

Creates or redefines calculated (symbolic) columns in a native table dictionary.

Syntax

Create_Symbolic (*tablename*, *columnname*, *formula*, *datatype*, *columnheading*, *multivalueflag*, *conversion*, *justification*, *length*, *description*)

Parameters

The Create_Symbolic routine has the following parameters.

Parameter	Description								
<i>tablename</i>	Name of table in which the calculated column is to appear.								
<i>columnname</i>	Specifies the name of the symbolic column to create or redefine. Column names must comply with rules for key values.								
<i>Formula</i>	The formula to execute when <i>columnname</i> is referenced. The formula is written using any valid BASIC+ expressions. The symbolic formula does not conform to rules defined for creating stored procedures. Note: Multiline formulae should be delimited by either @vm or @fm.								
<i>datatype</i>	The data type assigned to the column. If <i>datatype</i> is provided, the information associated with the data type, conversion, justification, and length, are automatically provided. If null, conversion, justification, and length must be provided in the appropriate arguments.								
<i>columnheading</i>	The heading for the symbolic column. If null, the default is <i>columnname</i> .								
<i>multivalueflag</i>	Specifies whether this column is single valued or multi-valued. If null, the default is single valued (S). Specify one of the codes listed below. <table><tr><th>Value</th><th>Description</th></tr><tr><td>S</td><td>Single valued.</td></tr><tr><td>M</td><td>Multi-valued.</td></tr></table>	Value	Description	S	Single valued.	M	Multi-valued.		
Value	Description								
S	Single valued.								
M	Multi-valued.								
<i>conversion</i>	The output format for the return data. If conversion and <i>datatype</i> are null, then no conversion is performed. If <i>datatype</i> is provided, the default is dependent on the data type assigned. See also, Oconv function.								
<i>justification</i>	The justification for the return data. If justification and <i>datatype</i> are null, the default is left (L) justified. If <i>datatype</i> is specified, justification defaults to the justification for the data type. Specify one of the values below. <table><tr><th>Value</th><th>Description</th></tr><tr><td>L</td><td>Left justified.</td></tr><tr><td>R</td><td>Right justified.</td></tr><tr><td>T</td><td>Left justified, text mark delimiters inserted at length intervals.</td></tr></table>	Value	Description	L	Left justified.	R	Right justified.	T	Left justified, text mark delimiters inserted at length intervals.
Value	Description								
L	Left justified.								
R	Right justified.								
T	Left justified, text mark delimiters inserted at length intervals.								
<i>length</i>	The length for the return data. If <i>length</i> and <i>datatype</i> are null, <i>length</i> defaults to a length of 20. If <i>datatype</i> is specified, <i>length</i> defaults to the length for the data type.								
<i>description</i>	Text describing the symbolic column.								

See also

[List_Dict](#)

Example

```
* This code example creates a symbolic field named "TAX" in the ORDERS table within the EXAMPLES application
Declare Subroutine Create_Symbolic, Set_Status, FsMsg
Declare Function Get_Status
table = "ORDERS"
column = "TAX"
formula = "@ans = ' ' : @vm : "@ans = {SUB_TOTAL} * .07" @fm
dType = "DOLLARS"
colHead = "Tax"
mvFlag = ""
conv = ""
just = "R"
length = 15
descr = "Calculate tax on Orders"
errCodes = ""
Set_Status(0)
Create_Symbolic(table, column, formula, dType, colHead, mvFlag, conv, just, length, descr)
If Get_Status(errCodes) Then
* Error Handling
FSMsg(errCodes)
end
```