m_utl_valid_name.sas File Reference

Utilities

Utility macro to convert a textual string into a valid SAS name

Description

The macro converts special characters found in text string into an underscore to comply to a SAS standard valid names. If the name is already valid, then the result will be the same as the input. Since name length in SAS is restricted to 32 characters, the result name is reduced automatically to 32 positions. The macro can be executed in the context of a SAS macro call or generates code to be executed in the context of a data step. This is controlled by the CONTEXT macro parameter. Valid values for CONTEXT are: _MACRO_ or _DATASTEP_.
The m_utl_valid_name.sas macro containes the following features:
For standard SAP BW names that start with /BIC/ or /BI0/, this
part of the text string will be removed. This is controlled by the macro parameter BW_SPECIAL. Valid values are: Y or N.
Macro parameter CHK_FIRST will check if the first character
of the text string is a / or a digit, or first character is a / followed by a digit. The first character of the name will be removed instead of being converted to an underscore. Valid values for CHK_FISRT are: Y or N.

Note

To be able to circumvent errors due to invalid table or variable names, the SAS system option VALIDVARNAME should be set to ANY in the SAS program that calls the m_utl_valid_name.sas SAS macro utility routine.

Autors

Paul Alexander Canals y Trocha (paul.canals@gmail.com)

Date

2020-09-07 00:00:00

Version

20.1.09

Link

https://github.com/paul-canals/toolbox

Parameters

Input	help	Parameter, if set (or ?) to print the Help information in the log. In all other cases this parameter should be left out from the macro call.
Input	in_var	Character string containing the input text string. The default value is: _NONE_.
Input	var	Alias of the _in_var_ parameter.
Input	out_var	Optional. This parameter is only valid when the CONTEXT parameter is set to DATASTEP and contains the name of the output variable. The default value is: _NONE_.
Input	outvar	Alias of the _out_var_ parameter.
Input	context	Indicator [DATASTEP|MACRO] to select the in which the macro is to be run in. The default value is: MACRO.
Input	bw_special	Boolean [Y|N] parameter to check standard SAP BW names that start with /BIC/ or /BI0/. If CHK_FIRST is set to Y then this part of the text string will be removed. The default value is: N.
Input	chk_first	Boolean [Y|N] parameter to check if the first character of the text string is a "/" or a digit, or first character is "/" followed by a digit. The default value is: Y.
Input	debug	Boolean [Y|N] parameter to provide verbose mode information. The default value is: N.

Returns

- Result valid SAS name.

Calls

- m_utl_print_message.sas
- m_utl_print_mtrace.sas

Usage

Example 1: Show help information:

```
%m_utl_valid_name(?)
```

Example 2: Macro context remove "0" to circumvent illegal first character:

```
%let varname=
  %m_utl_valid_name(
     var   = 0COUNTRY
   , debug = Y
     );

%put &=varname.;
```

Example 3: Macro context add underscore to circumvent illegal first character:

```
%let varname=
  %m_utl_valid_name(
     var       = 0COUNTRY
   , chk_first = N
   , debug     = Y
     );

%put &=varname.;
```

Example 4: Macro context no change between IN_VAR and VARNAME:

```
%let varname=
  %m_utl_valid_name(
     var   = NO_CHANGE
   , debug = Y
     );

%put &=varname.;
```

Example 5: Datastep context no change between INVAR and OUTVAR:

```
data WORK.result;
   invar="NO_CHANGE";
   %m_utl_valid_name(
      in_var  = invar
    , out_var = outvar
    , context = D
    , debug   = Y
      );
run;

proc print data=WORK.result;
run;
```

Example 6: Datastep context convert INVAR to valid OUTVAR with BW_SPECIAL=Y:

```
data WORK.result;
   invar="/BIC/BA_1FTRAN";
   %m_utl_valid_name(
      in_var     = invar
    , out_var    = outvar
    , context    = D
    , bw_special = Y
    , chk_first  = N
    , debug      = Y
      );
run;

proc print data=WORK.result;
run;
```

Example 7: Datastep context convert INVAR to valid OUTVAR with BW_SPECIAL=N:

```
data WORK.result;
   invar="/BIC/BA_1FTRAN";
   %m_utl_valid_name(
      in_var  = invar
    , out_var = outvar
    , context = D
    , debug   = Y
      );
run;

proc print data=WORK.result;
run;
```

Example 8: Datastep context convert /BIC/class to valid SAS table name:

```
data WORK.%m_utl_valid_name(var=%nrstr(/BIC/class),chk_first=N);
   set SASHELP.class;
run;

proc print data=WORK._BIC_class;
run;
```

Copyright