m_utl_get_col_code.sas File Reference

Utilities

Utility macro to obtain a data step or SQL style attribute list

---

Description

The macro is used to obtain a column attribute information list which is used to declare columns inside a data step, proc sql or database type sql. The attribute information may contain the column name, length, informat, format and label. The SHOW_ERR parameter shows or suppresses possible errors in the log. In case of a table with columns containing special attributes the VALID_NAME parameter converts these names to valid names. This paramater currently only works for proc SQL and DB type column name constructions. This macro is based on the ut_get_table_cols_code.sas macro by Dave Prinsloo (dave.prinsloo@yahoo.com).

Note

The SHOW_ERR parameter shows or suppresses possible warnings or errors in the log. The default for SHOW_ERR is value is N.

Note

In case of encrypted SAS datasets, the ENCRYPTKEY= parameter must be provided as part of the CREDS credentials string.

Autors

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

Date

2020-12-07 00:00:00

Version

20.1.12

Link

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

Parameters

Input	help	Parameter, if set (Help or ?) to print the Help information in the log. In all other cases this parameter should be left out from the macro call.
Input	table	Full LIBNAME.TABLENAME name of the or SAS dataset to get the column attribute information. The default value is: _NONE_.
Input	creds	Optional. Specifies the ENCRYPTKEY= parameter value if DATASET involves an encrypted dataset.
Input	valid_name	Boolean [Y|N] parameter to specify wether the result column names should be transformed to a valid SAS variable name containing no special characters. At present the parameter currently only works for DB and SQL type column attribute constructions. The default value is: N.
Input	type	Indicator [DATA|DB|ORACLE|SQL] to specify how the column attribute information list is constructed. The default value for TYPE is: DATA (Data Step).
Input	chartype	Indicator [BYTE|CHAR] to specify whether the Oracle character attribute type VARCHAR2 is to be declared as 'length' BYTE or CHAR. This argument is only valid for Oracle column type VARCHAR2. The default value is: BYTE.
Input	include	Optional. A blank separated list of columns to filter the result column list to only include columns listed in the include list parameter.
Input	exclude	Optional. A blank separated list of columns to filter the result column list to exclude columns listed in the exclude list parameter.
Input	sql_prefix	Optional. Parameter to specify the SQL type table alias name to be added as prefix to each column name in the result column attribute list.
Input	max_length	Optional. Parameter to specify the maximum length of a column in the result column attribute list.
Input	show_err	Boolean [Y|N] parameter to show or hide warnings or errors in the log. The default value is: Y.
Input	debug	Boolean [Y|N] parameter to provide verbose mode information. The default value is: N.

Returns

- List of data step or SQL style column attribute information.

Calls

- m_utl_get_col_attrib.sas
- m_utl_print_message.sas
- m_utl_print_mtrace.sas
- m_utl_varlist.sas

Usage

Example 1: Show help information:

```
%m_utl_get_col_code(?)
```

Example 2: Step 1 - Print original SASHELP.class table with missing labels:

```
proc print data=SASHELP.class(obs=1) noobs label;
run;
```

Example 2: Step 2 - Add the missing column label and format to the class table:

```
data WORK.class;
   attrib Name label='Name';
   attrib Sex label='Gender';
   attrib Age label='Age';
   attrib Height format=8.2 label='Height';
   attrib Weight format=8.2 label='Weight';
   set SASHELP.class (obs=1);
run;
```

Example 2: Step 3 - Create new table with added column attribute information:

```
data WORK.class2;
   attrib
   %m_utl_get_col_code(
      table = WORK.class
    , type  = DATA
    , debug = Y
      );
   set SASHELP.class (obs=1);
run;

proc print data=WORK.class noobs label;
run;
```

Copyright