m_utl_get_col_attrib.sas File Reference

Utilities

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

---

Description

The macro is used to obtain attribute information for a given column in a table. The macro is used to declare a column inside a data step, SAS proc SQL or database type SQL syntax. Column information may contain name, length, (in-)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_col_attrib.sas macro by Dave Prinsloo (dave.prinsloo@yahoo.com).

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	table_dsid	Parameter representing the SAS dataset or table identifier. The parameter contains the value of the table identifier when the table was already opened before calling this macro. The default value for TABLE_DSID is: 0.
Input	creds	Optional. Specifies the ENCRYPTKEY= parameter value if DATASET involves an encrypted dataset.
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	col_name	Parameter to specify the column attribute name.
Input	incl_name	Boolean [Y|N] parameter to specify if the result string includes the column name. The default value for INC_NAME is: Y.
Input	valid_name	Boolean [Y|N] parameter to specify wether the result column name should be transformed to a valid SAS variable name containing no special characters. At present the parameter currently only works for DB, ORACLE and SQL type column attribute constructions. The default value is: N.
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

- Data step or SQL style attribute information for a given column.

Calls

- m_utl_print_message.sas
- m_utl_print_mtrace.sas
- m_utl_valid_name.sas
- m_utl_varlist.sas

Usage

Example 1: Show help information:

```
%m_utl_get_col_attrib(?)
```

Example 2: Data step style - Add some column labels and formats to the attribute info:

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

data WORK.class;
   attrib
      %m_utl_get_col_attrib(table=SASHELP.class,col_name=Name,debug=Y) label='Student'
      %m_utl_get_col_attrib(table=SASHELP.class,col_name=Sex,debug=Y) label='Gender'
      %m_utl_get_col_attrib(table=SASHELP.class,col_name=Age,debug=Y) label='Age'
      %m_utl_get_col_attrib(table=SASHELP.class,col_name=Height,debug=Y) format=8.2
      %m_utl_get_col_attrib(table=SASHELP.class,col_name=Weight,debug=Y) format=8.2
      ;
   set SASHELP.class;
run;

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

Example 3: Proc SQL style - Add some column labels and formats to the attribute info:

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

proc sql noprint;
   create table WORK.class as
   select %m_utl_get_col_attrib(table=SASHELP.class,type=SQL,col_name=Name,debug=Y)
label='Student'
        , %m_utl_get_col_attrib(table=SASHELP.class,type=SQL,col_name=Sex,debug=Y)
label='Gender'
        , %m_utl_get_col_attrib(table=SASHELP.class,type=SQL,col_name=Age,debug=Y)
label='Age'
        , %m_utl_get_col_attrib(table=SASHELP.class,type=SQL,col_name=Height,debug=Y)
format=8.2
        , %m_utl_get_col_attrib(table=SASHELP.class,type=SQL,col_name=Weight,debug=Y)
format=8.2
    from SASHELP.class
   ;
quit;

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

Copyright