m_utl_save_hist_data.sas File Reference

Utilities

Utility macro to load new data into a historised SAS dataset

Description

This macro can be used to load new data into a historised table or dataset using a combination of valid date and version number as historisation attributes. With every load a new combination of valid date and version number is created, additionally also user and load timestamp values are written to the output table.

Note

SAS Problem Note 2859: LOCK statement or function with LIST or QUERY options might report locks incorrectly:
http://support.sas.com/kb/2/859.html

Autors

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

Date

2021-01-27 00:00:00

Version

21.1.01

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	in_tbl	Specifies the full LIBNAME.TABLENAME name of the source SAS dataset or database table. This parameter may contain a combination of data step style statement between brackets like keep=, drop= or rename=. The default value for IN_TBL is: _NONE_.
Input	intable	Alias of the IN_TBL= parameter.
Input	out_tbl	Specifies the full LIBNAME.TABLENAME name of the target SAS dataset or database table. The default value for OUT_TBL is: _NONE_.
Input	outtable	Alias of the OUT_TBL= parameter.
Input	where	Optional. Specifies a valid WHERE clause that selects observations from the IN_TBL SAS dataset. Using this argument subsets your data based on the criteria that you supply for the expression.
Input	valid_vars	Boolean [Y|N] parameter to specify wether the result column names should be transformed to a valid SAS variable name containing no special characters. The default value is: N.
Input	idcol	Specifies the name of load identifier column The default value for IDCOL is: LOAD_ID.
Input	dtcol	Specifies the name of load date column The default value for DTCOL is: LOAD_DT.
Input	datecol	Specifies the name of valid date selection column. The default value is: VALID_DT.
Input	valid_dt	Specifies the valid date value for loading data content. The value is to be specified by using the following format: _DD.MM.YYYY_.
Input	datecol	Specifies the name of valid date selection column. The default value is: VALID_DT.
Input	version	Optional. Specifies the of the valid date selection for filtering the data content. Do not use this unless you know what to do.
Input	verscol	Specifies the name of version selection column. The default value is: VERSION_NR.
Input	usercol	Specifies the name of user identifier column. The default value is: USER_ID.
Input	loadcol	Specifies the name of loading datetime column. The default value is: LOAD_DTTM.
Input	type_cd	Optional. Specifies the result type code selection for filtering the data content.
Input	typecol	Specifies the name of result type selection column. The default value is: RESULT_CD.
Input	timeout	Specifies the waiting time in seconds to wait until a lock can be set on the output SAS dataset. The default value is: 300.
Input	debug	Boolean [Y|N] parameter to provide verbose mode information. The default value is: N.

Returns

- Loaded data into a historised SAS dataset output table.

Calls

- m_utl_chk_reqcols.sas
- m_utl_clr_table_lock.sas
- m_utl_create_index.sas
- m_utl_delete_index.sas
- m_utl_get_col_code.sas
- m_utl_get_userid.sas
- m_utl_print_message.sas
- m_utl_print_mtrace.sas
- m_utl_nlobs.sas
- m_utl_set_table_lock.sas

Usage

Example 1: Show help information:

```
%m_utl_save_hist_data(?)
```

Example 2: Load data from SASHELP.classfit into a historised table (no LOAD_CD):

```
%* Initialize classfit table ;
proc datasets lib=WORK nolist nowarn memtype=(data view);
   delete classfit;
quit;

%m_utl_save_hist_data(
   in_tbl   = SASHELP.classfit (drop=Sex rename=(Name=Student))
 , out_tbl  = WORK.classfit
 , where    = %str(Sex = 'F' and Age > 12)
 , valid_dt = 30.09.2017
 , debug    = Y
   );

%m_utl_save_hist_data(
   intable  = SASHELP.classfit (drop=Sex rename=(Name=Student))
 , outtable = WORK.classfit
 , where    = %str(Sex = 'M' and Age > 12)
 , valid_dt = 30.09.2017
 , debug    = Y
   );

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

Example 3: Load data from SASHELP.class into a historised table (with LOAD_CD):

```
%* Initialize class table ;
proc datasets lib=WORK nolist nowarn memtype=(data view);
   delete class;
quit;

%m_utl_save_hist_data(
   in_tbl   = SASHELP.class
 , out_tbl  = WORK.class
 , where    = %str(Sex = 'F')
 , valid_dt = 30.09.2017
 , verscol  = version
 , type_cd  = GIRLS
 , debug    = Y
   );

%m_utl_save_hist_data(
   intable  = SASHELP.class
 , outtable = WORK.class
 , where    = %str(Sex = 'M')
 , valid_dt = 30.09.2017
 , verscol  = version
 , type_cd  = BOYS
 , debug    = Y
   );

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

Copyright