

m_utl_drop_column.sas File Reference

Utilities

Utility macro to drop a column from a table or dataset

Description

The macro tries to drop a column from an existing database table or SAS dataset. The column name is checked against the table or dataset on its existence.

Note

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

Authors

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

Date

2023-07-27 00:00:00

Version

23.1.07

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	dblib	SAS dataset library or database libref name.
Input	dbtbl	SAS dataset or database table name.
Input	dbtyp	Indicator [DB2 ODBC ORA SAS] to specify the database type. The default value is: SAS.
Input	colnm	The column Name. The maximum length is 30.
Input	creds	String containing given database credentials containing user=, password=, path= or datasrc= and schema= information. The following order of parameters is to be respected for the macro to work correctly: DB2: user=, password=, path=, schema= ODBC: user=, password=, dsn=, schema= ORA: user=, password=, path=, schema= SAS: (Optional) encryptkey=
Input	print	Boolean [Y N] parameter to generate the output by a proc report step with style HtmlBlue. The default value is: N.
Input	debug	Boolean [Y N] parameter to provide verbose mode information. The default value is: N.

Returns

- A column dropped from an existing database table

Calls

- [m_log_set_options.sas](#)
- [m_utl_print_message.sas](#)
- [m_utl_print_mtrace.sas](#)
- [m_utl_unique_number.sas](#)

Usage

Example 1: Show help information:

```
%m_utl_drop_column(?)
```

Example 2: Drop a column from an encrypted SAS dataset:

```
data WORK.class(encrypt=aes encryptkey=aespasskey);
  set SASHELP.class;
run;

proc print data=WORK.class(encryptkey=aespasskey);
run;

%m_utl_drop_column(
  dblib = WORK
  , dbtbl = class
  , dbtyp = SAS
  , colnm = Weight
  , creds = %str(encryptkey=aespasskey)
  , debug = Y
  );

proc print data=WORK.class(encryptkey=aespasskey);
run;
```

Example 3: Drop a column from an Oracle table:

```
*libname ORADB oracle user=orademo pwd='ORApw123' path=XE schema=orademo;

*proc datasets lib=ORADB nolist nowarn;
*  delete class;
*quit;

*data ORADB.class;
*  set SASHELP.class;
*run;

*%m_utl_drop_column(
*  dblib = ORADB
*  , dbtbl = CLASS
*  , dbtyp = ORA
*  , colnm = HEIGHT
*  , creds = %str(user=orademo pwd=ORApw123 path=XE schema=orademo)
*  , debug = Y
*  );
```

Copyright

Copyright 2008-2023 Paul Alexander Canals y Trocha.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.