

# m\_utl\_add\_column.sas File Reference

## Utilities

Utility macro to add a column to an existing table or dataset

---

### Description

The macro tries to add a column to an existing database table or SAS dataset. The given column name is checked against a list of database reserved words, the given column type and length. The macro was originally designed for adding columns to an Oracle table, but has been expanded ever since.

### Note

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

### Autors

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	coltp	Indicator [BIN CHAR DATE NUM] to select the column type. The default value is CHAR.
Input	chartp	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	colln	The column length. The maximum length is 4000. This value relevant for the CHAR and NUM types. If NUM then length may also contain precision information. For example "6.2".
Input	colfm	Optional for SAS type only. The column format. This value relevant for the DATE and NUM types. If NUM then format may also contain precision information.   For example: 8.2 , date9. , yyyymmddp10.
Input	creds	String containing given database credentials containing user=, password=, path= and schema= information. The following order of parameters is to be respected for the macro to work;   DB2: user=, password=, path=, 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 new column to an existing database table

## Calls

- [m\\_log\\_set\\_options.sas](#)
- [m\\_utl\\_chk\\_reserved\\_words.sas](#)
- [m\\_utl\\_print\\_message.sas](#)
- [m\\_utl\\_print\\_mtrace.sas](#)
- [m\\_utl\\_unique\\_number.sas](#)

## Usage

Example 1: Show help information:

```
%m_utl_add_column(?)
```

Example 2: Add a numerical column to an encrypted SAS dataset:

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

%m_utl_add_column(
    dblib = WORK
    , dbtbl = class
    , dbtyp = SAS
    , colnm = Income
    , coltp = NUM
    , colln = 8
    , colfm = 8.2
    , creds = %str(encryptkey=aespasskey)
    , debug = Y
    );

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

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

Example 3: Add a numerical column to 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_add_column(
*  dblib = ORADB
*  , dbtbl = CLASS
*  , dbtyp = ORA
*  , colnm = RECNUM
*  , coltp = NUM
*  , colln = 8.2
*  , 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/>>.