

# m\_utl\_table\_size.sas File Reference

## Utilities

Utility macro to determine the uncompressed size of a table

---

### Description

The macro can be used anywhere in a SAS program including within a SAS procedure or SAS data step. If the table does not exist, it returns -1. If the table exists, but cannot be opened it returns 0. Different parameter names are allowed. This macro is based on the sizedata.sas macro program by Michael A. Raithel (michaelraithel1@verizon.net).

### 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.*

### Authors

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

### Date

2020-09-07 00:00:00

### Version

20.1.09

### Link

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

## Parameters

Input	help	Parameter, if set ( or ?) to print the Help information in the log. In all other cases this parameter should be left out from the macro call.
Input	dataset	Full LIBNAME.TABLENAME name of the table or SAS dataset to get the number of records from. The parameter can contain SAS data step style where statement between brackets. See examples 3 and 4. The default value is: <code>_NONE_</code> .
Input	table	Alias of the dataset= parameter.
Input	data	Alias of the dataset= parameter.
Input	creds	Optional. Specifies the ENCRYPTKEY= parameter value if DATASET involves an encrypted dataset.
Input	bufsize	Specifies the size of a permanent buffer page for an output dataset, defined by the operating system dependent system option BUFSIZE. The initial value of the macro argument is checked against the system option. If the system option value is equal to 0, the default value of 65536 is used. If the BUFSIZE is set to 0, the calculated table size is defined by the dictionary tables information. If the file size is not found the LRECL*NOBS formula is used.
Input	nrecs	Optional. Specifies the number of records for the table size calculation. If the NRECS parameter is not provided, the macro obtains the number of records using the <code>m_utl_nlobs.sas</code> macro.
Input	format	Boolean [Y N] parameter to specify if the result size is to be formatted by the SIZEKMG format. See examples 4 and 5. The default value is: N.
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

- The table size of a table or SAS dataset in bytes

## Calls

- [m\\_utl\\_get\\_attrn.sas](#)
- [m\\_utl\\_nlobs.sas](#)
- [m\\_utl\\_print\\_message.sas](#)
- [m\\_utl\\_print\\_mtrace.sas](#)

## Usage

Example 1: Show help information:

```
%m_utl_table_size(?)
```

Example 2: Calculate the table size from an encrypted SAS dataset:

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

proc print data=WORK.class(encryptkey=aespasskey);
  where Sex='F';
run;

%let table_size=
  %m_utl_table_size(
    table = WORK.class(where=(Sex='F'))
    , creds = %str(encryptkey=aespasskey)
    , debug = Y
  );

%put Table size of WORK.class is: &table_size. bytes;

data WORK.size;
  table='WORK.class';
  sex='F';
  size=&table_size.;
run;

proc print data=WORK.size noobs;
run;
```

Example 3: Calculate the table size from a SAS dataset with where statement:

```
data WORK.size;
  attrib table length=$32. label='Table';
  attrib size length=8 format=sizekmg12.1;
  table='SASHELP.class';
  size=%m_utl_table_size(
    table = SASHELP.class(where=(Age > 13))
    , debug = Y
  );
run;

proc print data=WORK.size noobs;
run;
```

Example 4: Calculate the formatted table size from a SAS dataset with where statement:

```
data WORK.size;
  table='SASHELP.class';
  size="%m_utl_table_size(
    table = SASHELP.class(where=(Age > 13))
    , format = Y
    , debug = Y
  )";
run;

proc print data=WORK.size noobs;
run;
```

Example 5: Calculate the table size for a table with 5.000.000 records:

```
%let table_size=
  %m_utl_table_size(
    data    = SASHELP.class
    , nrecs  = 5000000
    , format = Y
    , debug  = Y
  );

%put Table size for SASHELP.class with 5M records is: &table_size.;
```

## **Copyright**

Copyright 2008-2020 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/>>.