

# m\_utl\_copy\_sas\_file.sas File Reference

## Utilities

Utility macro to copy a SAS data type files to another location

---

### Description

This program is capable of copying datasets, indexes, catalogs and programs onto another location. In case of cross platforms with different formats or encoding the copying is handled by the SAS Cross Environment Data Access (CEDA) utility.

### Note

*This program is able to work in system environments where x-command or unix pipes are not allowed or cannot be used.*

### Autors

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

### Date

2020-03-02 00:00:00

### Version

20.1.03

### 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 | indir  | Full qualified path to the input directory.  |
| Input | outdir | Full qualified path to the output directory.   |
| Input | inlib  | Optional. Full name or libref of the input library. The default value is: _INLIB_.   |
| Input | outlib | Optional. Full name or libref of the output library. The default value is: _OUTLIB_.   |
| Input | clone  | Boolean [Y N] parameter to make a copy of the original file. The encoding, data representation and other options of the original file are copied to the clone file. This may have consequences when data is copied between different hosts and operating systems. The default value is: Y. |
| Input | select | Optional. Parameter identify items to be copied to the output directory, separated by a blank. Default value is _ALL_ to select all items.   |
| Input | enckey | Optional. The encryption key, case when one or more input dataset are encrypted. The encryption key needs to be identical over the datasets.   |
| Input | debug  | Boolean [Y N] parameter to provide verbose mode information. The default value is: N.  |

## Returns

- The copied SAS files as mentioned in OUTDIR or OUTLIB.

## Calls

- [m\\_utl\\_create\\_dir.sas](#)
- [m\\_utl\\_print\\_message.sas](#)

## Usage

Example 1: Show help information:

```
%m_utl_copy_sas_file(?)
```

Example 2: Copy SASHELP.class to an output directory:

```
%m_utl_copy_sas_file(  
  inlib = SASHELP  
  , outdir = %sysfunc(getoption(WORK))  
  , select = class  
  , debug = Y  
);
```

Example 3: Copy a SAS catalog to an output directory:

```
options dlcreatedir;  
libname TEMP "%sysfunc(getoption(WORK))/backup";  
options nodlcreatedir;  
  
%m_utl_copy_sas_file(  
  indir = %sysfunc(pathname(SASHOME))/SASFoundation/9.4/core/macros  
  , outlib = TEMP  
  , select = sasmacr.sas7bcat  
  , debug = Y  
);  
  
proc catalog cat=TEMP.sasmacr;  
  contents;  
  run;  
quit;  
  
libname TEMP clear;
```

Example 4: Copy a dataset and its index to another library:

```
data WORK.class;  
  set SASHELP.class;  
run;  
  
proc datasets lib=WORK nodetails nolist;  
  modify class;  
  index create name /unique;  
  run;  
quit;  
  
proc contents data=WORK.class;  
run;  
  
options dlcreatedir;  
libname TEMP "%sysfunc(getoption(WORK))/backup";  
options nodlcreatedir;  
  
%m_utl_copy_sas_file(  
  inlib = WORK  
  , outlib = TEMP  
  , select = class  
  , debug = Y  
);  
  
proc contents data=TEMP.class;  
run;  
  
libname TEMP clear;
```

Example 5: Copy an encrypted dataset to another library:

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

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

options dlcreatedir;
libname TEMP "%sysfunc(getoption(WORK))/backup";
options nodlcreatedir;

%m_utl_copy_sas_file(
    inlib = WORK
    , outlib = TEMP
    , select = class
    , enckey = aespasskey
    , debug = Y
    );

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

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

libname TEMP clear;
```

## **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/>>.