m_utl_printto.sas File Reference

Utilities

Utility macro to set the SAS proc printto log output location

---

Description

This program sets the SAS proc printto log output destination to either an external log file or to the default log window. If the log output destination was already set to an external file, a printto list will be created so that the log output destination can be set back to its former state. This also works for nested printto log file statements. A notification with the full printto list of will be printed in the log file.

Autors

       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	log_dest	Indicator [F|FILE|L|LOG] to specify the output log destination. If the parameter value is set to L or LOG, the output is set to the default log window destination. If the parameter value is set to F or FILE, the output is set to the output location set by the LOG_FILE parameter. The default value for LOG_DEST is: _NONE_.
Input	log_file	Specifies the directory path and name of the log file where the log output is to be written into. If the specified file does not exist it will be created, otherwise overwritten by default. The default value for LOG_FILE is: _NONE_.
Input	new_file	Optional. Boolean [Y|N] Parameter to specify wether to create a new log file, or to append to an already existing log file. The default value for NEW_FILE is: Y.
Input	prefix	Optional. Specifies a string added to the name of the output log file.
Input	sdir_flg	Boolean [Y|N] parameter to specify whether sub directories are created to extend the log path containing year, month and day information. The default value for SDIR_FLG is: N.
Input	sdir_fmt	Optional. INdicator [Y|YM|YMD] to specify the format used to generate the sub directories. The default value for SDIR_FMT is: YMD.
Input	timestamp	Optional. Boolean [Y|N] parameter to specify whether timstamp information (yyyymmdd_hhmmss) is added to the name of the output log file. The default value for TIMESTAMP is: N.
Input	global_flg	Boolean [Y|N] Parameter to specify wether the MVAR_NAME parameter value is to be declared as a global macro variable. If the parameter value is set to N, MVAR_NAME is not returned outside the macro. The default value is: N.
Output	mvar_name	Name of the global macro variable containing the result value representing the log output destination. The default value is: _printto.
Input	debug	Boolean [Y|N] parameter to provide verbose mode information. The default value is: N.

Returns

- Destination output to SAS default LOG or external log file.

Calls

- m_utl_create_dir.sas
- m_utl_print_message.sas
- m_utl_print_mtrace.sas

Usage

Example 1: Show help information:

```
%m_utl_printto(?)
```

Example 2: Set the proc printto log output destination to a file:

```
%m_utl_printto(
   log_dest = FILE
 , log_file = %str(%sysfunc(getoption(WORK))/output.log)
 , debug    = Y
   );

data WORK.class;
   set SASHELP.class;
run;

%m_utl_printto(
   log_dest = LOG
 , debug    = Y
   );

%put PRINT LOG_FILE:;

filename log_file "%sysfunc(getoption(WORK))/output.log";

data _null_;
   infile log_file;
   input;
   put _infile_;
run;
```

Example 3: Set the proc printto log output to a file with prefix:

```
%m_utl_printto(
   log_dest = FILE
 , log_file = %str(%sysfunc(getoption(WORK))/output.log)
 , prefix   = Test
 , debug    = Y
   );

data WORK.class;
   set SASHELP.class;
run;

%m_utl_printto(
   log_dest = LOG
 , debug    = Y
   );

%put PRINT LOG_FILE:;

filename log_file "%sysfunc(getoption(WORK))/test_output.log";

data _null_;
   infile log_file;
   input;
   put _infile_;
run;
```

Example 4: Set the proc printto log output to a file with timestamp in name:

```
%m_utl_printto(
   log_dest   = FILE
 , log_file   = %str(%sysfunc(getoption(WORK))/output.log)
 , timestamp  = Y
 , global_flg = Y
 , mvar_name  = _printto_file
 , debug      = Y
   );

data WORK.class;
   set SASHELP.class;
run;

%m_utl_printto(
   log_dest = LOG
 , debug    = Y
   );

%put PRINT LOG_FILE:;

filename log_file "&_printto_file.";

data _null_;
   infile log_file;
   input;
   put _infile_;
run;
```

Example 5: Set nested proc printto log outputs to multiple log files:

```
%m_utl_printto(
   log_dest = FILE
 , log_file = %sysfunc(getoption(WORK))/log1.log
 , debug    = Y
   );
%put &_sysprinttolist.;

data WORK.class;
   set SASHELP.class;
run;

%m_utl_printto(
   log_dest = FILE
 , log_file = %sysfunc(getoption(WORK))/log2.log
 , debug    = Y
   );
%put &_sysprinttolist.;

data WORK.classfit;
   set SASHELP.classfit;
run;

%m_utl_printto(
   log_dest = FILE
 , log_file = %sysfunc(getoption(WORK))/log3.log
 , debug    = Y
   );
%put &_sysprinttolist.;

%m_utl_printto(
   log_dest = LOG
 , debug    = Y
   );

proc print data=WORK.classfit;
run;

%m_utl_printto(
   log_dest = LOG
 , debug    = Y
   );

proc print data=WORK.class;
run;

%m_utl_printto(
   log_dest = LOG
 , debug    = Y
   );

filename log_file "%sysfunc(getoption(WORK))/log1.log";
data _null_;
   infile log_file;
   input;
   put 'PRINT LOG1:' _infile_;
run;

filename log_file "%sysfunc(getoption(WORK))/log2.log";
data _null_;
   infile log_file;
   input;
   put 'PRINT LOG2:' _infile_;
run;

filename log_file "%sysfunc(getoption(WORK))/log3.log";
data _null_;
   infile log_file;
   input;
   put 'PRINT LOG3:' _infile_;
run;
```

Copyright