

m_log_run_analysis.sas File Reference

Logging

Logging macro to build a log analysis summary from a log file

Description

This program is used to perform overall runtime performance on SAS steps and run status analysis, on a given SAS log file. The result information is presented in the result tab by using a SAS proc report step, including a hyperlink to download the log file including line number at the beginning of each line. Optionally the macro can run SAS data step performance analyses to determine the run time, the size of the input table, and the size of the output table written into a (temporary) log file. This macro is originally based on the experimental SAS logparse macro paper by Michael A. Raithel (michaelraithel1@verizon.net)

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	in_file	Specifies the name and location of the SAS log file. The default value for IN_FILE is: _NONE_.
Input	program	Optional. The name of the SAS program, macro routine or SAS Stored Process that was run to which the given log file (IN_FILE) relates to.
Input	run_mode	Run mode parameter to specify the analysis run type. The valid values for RUN_MODE are LOG and COPY. If LOG, the macro analyses a given logfile. If COPY, the macro executes a SAS datastep copy and performs performance analysis on the SAS data step copy. The default value for RUN_MODE is: LOG.
Input	in_tbl	Optional. Full qualified name <library.table>. The default value for IN_TBL is: _NONE_.
Input	out_tbl	Optional. Full qualified name <library.table>. The default value for OUT_TBL is: _NONE_.
Input	file_flg	Optional. Boolean [Y N] parameter to specify whether the output table or file size is to be calculated. The default value for FILE_FLG is: N.
Input	print	Boolean [Y N] parameter to generate the output by using proc report steps with style HtmlBlue. The default value for PRINT is: N.
Input	print_sum	Boolean [Y N] parameter to generate the summary output. The default value for PRINT_SUM is: Y.
Input	print_log	Boolean [Y N] parameter to specify if a detailed log is to be added to the ODS output (PDF/HTML). The default value for PRINT_LOG is: N.
Input	print_pdf	Boolean [Y N] parameter to generate the analysis and summary output into an external pdf file. The default value for PRINT_PDF is: N.
Input	print_htm	Boolean [Y N] parameter to generate the analysis and summary output into an external html file. The default value for PRINT_HTM is: N.
Input	debug	Boolean [Y N] parameter to provide verbose mode information. The default value is: N.

Returns

- Table containing summarized log analysis results.

Calls

- [m_log_set_options.sas](#)
- [m_util_chk_table_exist.sas](#)
- [m_util_finfo_size.sas](#)
- [m_util_nlobs.sas](#)
- [m_util_print_message.sas](#)
- [m_util_print_mtrace.sas](#)
- [m_util_printto.sas](#)
- [m_util_table_size.sas](#)
- [m_util_unique_number.sas](#)

Usage

Example 1: Show help information:

```
%m_log_run_analysis(?)
```

Example 2: Step 1 - Set the log destination to a log file:

```
options fullstimer;

%let _sysprinttolog = %sysfunc(tranwrd(%nrquote(&sysprinttolog.),%str(%),%str()));
%put &_sysprinttolog.;

filename tmpfile "%sysfunc(getoption(work))/sas.log";

proc printto log=tmpfile new;
run;
```

Example 2: Step 2 - Run some data steps and procedures:

```
data WORK.blaat;
  set SASHELP.class;
  where John eq 'John';
run;

proc sql noprint;
  create table WORK.cars as
  select *
  from SASHELP.cars
  ;
quit;

proc sort data=SASHELP.prdsale
  out=WORK.prdsale;
  by descending year month;
run;
```

Example 2: Step 3 - Return to the default log destination:

```
proc printto log=%sysfunc(ifc(%length(%nrquote(&_sysprinttolog.)) eq 0,
  log,"%_sysprinttolog.");
run;

filename tmpfile clear;

options nofullstimer;
```

Example 2: Step 4 - Reset return code and syntaxcheck options:

```
%let syscc = 0;

options nosyntaxcheck obs=max;
```

Example 2: Step 5 - Perform runtime analysis on sas.log file:

```
%m_log_run_analysis(
  in_file   = %str(%sysfunc(getoption(work))/sas.log)
  , program = LOG_ANALYSIS
  , print   = Y
  , print_sum = Y
  , print_pdf = Y
  , print_htm = Y
  , debug    = N
);
```

Example 3: Perform performance analysis on a data step copy:

```
%m_log_run_analysis(  
  in_file      = %str(%sysfunc(getoption(work))/sas.log)  
, program     = RUN_ANALYSIS  
, run_mode    = COPY  
, in_tbl      = SASHELP.cars  
, out_tbl     = WORK.cars  
, file_flg    = Y  
, print       = Y  
, print_log   = Y  
, print_pdf   = Y  
, print_htm   = Y  
, debug       = N  
);
```

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/>>.