m_adm_compile_functs.sas File Reference

Admin

Admin macro to compile functions into a library container

Description

The macro lists all SAS functions from a given directory and adds these into a function library container using compilation. The result information is presented by a SAS Proc Report step step and can be send by email as an PDF format attachment.

Note

To guarantee a successful creation of the function library container included with the function source files, each of the PROC FCMP OUTLIB calls in the function source files should always refer to the macro variable name: "&OUTLIB"!
(e.g. PROC FCMP OUTLIB=&OUTLIB.;)

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 (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	indir	Specifies the full path to the input directory containing the SAS functions program files. The default value is: _NONE_.
Input	outdir	Specifies the full path to the output function library container. The default value is: _NONE_.
Input	outlib	Specifies the three level name for the function library container. The default value is: _NONE_.
Input	print	Boolean [Y|N] parameter to generate the output by a proc report step with style HtmlBlue. The default value is: Y.
Input	sendmail	Boolean [Y|N] parameter to decide if the result information list is to be send to email address. The default value is: N.
Input	mailaddr	Send to email address of private person or group.
Input	debug	Boolean [Y|N] parameter to provide verbose mode information. The default value is: N.

Returns

- WORK.USER_GROUPS

Calls

- m_utl_create_dir.sas
- m_utl_get_file_list.sas
- m_utl_mail_notification.sas
- m_utl_nlobs.sas
- m_utl_print_message.sas
- m_utl_print_mtrace.sas

Usage

Example 1: Show help information:

```
%m_adm_compile_functs(?)
```

Example 2 - Step 1: Create a functions using SAS proc fcmp syntax:

```
options dlcreatedir;
libname mylib "%sysfunc(getoption(WORK))/functions";
filename myfile "%sysfunc(getoption(WORK))/functions/translate_to_spanish.sas";

data _null_;
   file myfile;
   put '* proc fcmp outlib=TEMP.funcs.maths; ;' ;
   put '   function translate_to_spanish(x $) $ 12;' ;
   put '      if x = "yes" then return("si");' ;
   put '      else return("no");' ;
   put '   endsub;' ;
   put '* quit; ;' ;
run;

filename myfile clear;
libname mylib clear;
options nodlcreatedir;
```

Example 2 - Step 2: Compile functions to a library container and print report:

```
%m_adm_compile_functs(
   indir    = %sysfunc(getoption(WORK))/functions
 , outdir   = %sysfunc(getoption(WORK))/catalog
 , outlib   = TEMP.functs.samples
 , print    = Y
 , debug    = Y
   );

data WORK.test;
   english='yes';
   spanish=translate_to_spanish('yes');
   put spanish=;
run;

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

Example 3 - Step 3: Compile functions to a library container and email report:

```
%m_adm_compile_functs(
   indir    = %sysfunc(getoption(WORK))/functions
 , outdir   = %sysfunc(getoption(WORK))/catalog
 , outlib   = TEMP.functs.samples
 , sendmail = Y
 , mailaddr = %str(pact@hermes.local)
 , debug    = Y
   );
```

Copyright