

m_cst_set_parameter.sas File Reference

Custom

Custom macro to register global macro variables from a file

Description

This program reads a list of macro variables defined in an input CSV file or SAS dataset. Records in the file can either be environment dependent (INHOST) or not, or set inactive in case of PRMFLG set to 0. The following macro variable types are valid:

CHAR : Character value(s list)

DATE : Uses DATE format on date() input, returns numerical

value on DDMONYYYY input (e.g.\ 31dec2016) DTTM : Uses DATETIME format on datetime() input, returns

numerical value on DDMONYYYY HH:MM:SS input

(e.g.\ 31dec2016 23:59:59)

FUNC : Uses %sysfunc() function

MVAR : Uses %str(&)PRMVAL%str(.

NUM : Numerical value(s list)

SGET : Uses %sysget() function

Note

It is also possible to create macro variables containing a list of values. This is valid for CHAR and NUM type macro variables only. The key is to name the PRMVAR macro variable identically (e.g.\ list the values in the list as records in the INFILE file or dataset).

Autors

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

Date

2023-09-11 00:00:00

Version

23.1.09

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	infile	Full qualified path to the parameter file, or full LIBNAME.TABLENAME name of the SAS dataset.
Input	intype	Indicator [CSV TBL] for the INFILE type. The default value is: CSV.
Input	inhost	Indicator [DEV TST UAT PRD] for the environment. If omitted, the records in the INFILE with PRMENV value ALL will be set, all other will be ignored.
Input	prmenv	Name of the parameter environment column. The Default value is: PARAM_ENV.
Input	prmvar	Name of the parameter name column. The Default value is: PARAM_NAME.
Input	prmval	Name of the parameter value column. The Default value is: PARAM_VALUE.
Input	prmtyp	Name of the parameter type column. The Default value is: PARAM_TYPE.
Input	prmtxt	Name of the parameter description column. The Default value is: PARAM_TEXT.
Input	prmflg	Name of the parameter active/inactive flag column. The Default value is: MACRO_FLG.
Input	debug	Boolean [Y N] parameter to provide verbose mode information. The default value is: N.

Returns

- List of globally declared SAS macro variables

Calls

- [None](#)

Usage

Example 1: Show help information:

```
%m_cst_set_parameter(?)
```

Example 2 - Step 1: Create macro variable parameter table:

```
data WORK.params;
  attrib PARAM_ENV    length = $3;
  attrib PARAM_NAME   length = $32;
  attrib PARAM_VALUE  length = $100;
  attrib PARAM_TYPE   length = $4;
  attrib PARAM_TEXT   length = $100;
  attrib MACRO_FLG    length = 8;

  * Date Function;

  PARAM_ENV    = "ALL";
  PARAM_NAME   = "X_TODAY_FUNC";
  PARAM_VALUE  = "date()";
  PARAM_TYPE   = "FUNC";
  PARAM_TEXT   = "Represents today in numerical string";
  MACRO_FLG    = 1;
  output;

  * Date Type with date() input;

  PARAM_ENV    = "ALL";
  PARAM_NAME   = "X_TODAY_DATE";
  PARAM_VALUE  = "date()";
  PARAM_TYPE   = "DATE";
  PARAM_TEXT   = "Represents today in date format";
  MACRO_FLG    = 1;
  output;

  * Date Type with 31dec2016 input;

  PARAM_ENV    = "ALL";
  PARAM_NAME   = "X_DAY_DATE";
  PARAM_VALUE  = "31dec2016";
  PARAM_TYPE   = "DATE";
  PARAM_TEXT   = "Represents today in date format";
  MACRO_FLG    = 1;
  output;

  * List Character Type;

  PARAM_ENV    = "ALL";
  PARAM_NAME   = "X_WEEKEND_LIST";
  PARAM_VALUE  = "Saturday";
  PARAM_TYPE   = "CHAR";
  PARAM_TEXT   = "Represents the weekend day names";
  MACRO_FLG    = 1;
  output;

  PARAM_ENV    = "ALL";
  PARAM_NAME   = "X_WEEKEND_LIST";
  PARAM_VALUE  = "Sunday";
  PARAM_TYPE   = "CHAR";
  PARAM_TEXT   = "Represents the weekend day names";
  MACRO_FLG    = 1;
  output;
run;
```

Example 2 - Step 2: Register macro variables from params table:

```
%m_cst_set_parameter(  
    infile = WORK.params  
    , intype = TBL  
    , inhost =  
    , prmenv = PARAM_ENV  
    , prmvar = PARAM_NAME  
    , prmval = PARAM_VALUE  
    , prmtyp = PARAM_TYPE  
    , prmtxt = PARAM_TEXT  
    , prmflg = MACRO_FLG  
    , debug  = N  
    );
```

Example 2 - Step 3: Check registered macro variables in log:

```
%put &=X_TODAY_FUNC.;  
%put &=X_TODAY_DATE.;  
%put &=X_DAY_DATE.;  
%put &=X_WEEKEND_LIST.;
```

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/>>.