m_utl_chk_func_exist.sas File Reference

Utilities

Utility macro to check if a user defined function is registered

---

Description

This macro checks wether a user defined function is registered in a given function library container. If the function is found in the function library container, the value of the output macro variable MVAR_MATCH is set to 1, otherwise the value is set to 0.

Note

This macro can be used as inline code by setting the parameter GLOBAL_FLG to N. In this case the MVAR_MATCH variable is set as a local SAS macro variable.

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	fname	Full name of the user defined function to check. The default value is: _NONE.
Input	function	Alias of the FNAME= parameter.
Input	flib	Full LIBNAME.TABLENAME name of the function library and container. The default value is: _NONE.
Input	library	Alias of the FLIB= parameter.
Input	global_flg	Boolean [Y|N] Parameter to specify wether the result value is to be declared as a global macro variable. If set to N, the result is only returned inline. The default value for GLOBAL_FLG is: N.
Output	mvar_match	Name of the global SAS macro variable containing a boolean [0|1] expression value representing the result of the function match query. The default value for MVAR_MATCH is: _f_match.
Input	msg_type	Determines the severity of the message in case the function could not be found in the given library. The default value for MSG_TYPE is: ERR.
Input	debug	Boolean [Y|N] parameter to provide verbose mode information. The default value is: N.

Returns

- Returns a 1 or 0 depending on function match.

Calls

- m_utl_print_message.sas
- m_utl_print_mtrace.sas

Usage

Example 1: Show help information:

```
%m_utl_chk_func_exist(?)
```

For the next examples create a function library and functions:

```
options cmplib=WORK.functs;

proc fcmp outlib=WORK.functs.examples;
   function year_of_birth(age);
      birth_year = year(date())-age;
      return(birth_year);
   endsub;
quit;
```

Example 2: check if the function Day_of_Birth exists (Result=0):

```
%let function_exist =
   %m_utl_chk_func_exist(
      fname = Day_of_Birth
    , flib  = WORK.functs
    , debug = Y
      );

%put &=function_exist.;
```

Example 3: check if the function Month_of_Birth exists (Result=0):

```
%m_utl_chk_func_exist(
   fname      = Month_of_birth
 , flib       = WORK.functs
 , global_flg = Y
 , mvar_match = func_exist
 , debug      = Y
   );

%put &=func_exist.;
```

Example 4: check if the function Year_of_Birth exists (Result=1):

```
%m_utl_chk_func_exist(
   function   = Year_of_birth
 , library    = WORK.functs
 , global_flg = Y
 , mvar_match = func_exist
 , debug      = Y
   );

%put &=func_exist.;
```

Copyright