

m_sys_upd_dbaccess.sas File Reference

System

System macro to (re-)activate a dbms or SAS libref connection

Description

This macro sets a certain database connection into an active or inactive state. This macro is either called from the admin SAS stored proces, or from the m_sys_set_dbaccess.sas macro program when the user password combination is invalid during logon to a database connection. The program creates a backup of the DBACCESS table before making any changes to the ACTIVE state of an entry.

Note

The SAS dataset containing the database connection profiles has to be named DBACCESS since all the routines expects this.

Note

If you lose or forget the ENCRYPTKEY, there will be absolutely no way to open the DBACCESS table and recover the data!

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	dstable	Full qualified table name <LIBNAME.dbaccess>.
Input	dskey	Value of the ENCRYPTKEY parameter. The minimum string length needs to be 8 characters long.
Input	dname	Name of the database connection or SAS libref. The maximum length is 8 characters.
Input	duser	Database access user name.
Input	daccs	Mandatory. Indicator [P T] for the entry access mode user type (DUSR) which can be (P)ersonal or (T)echnical. This parameter must always be set, even if the database type (DTYP) is set to SAS. The default value is: T (technical).
Input	ddact	Boolean [Y N] flag to indicate that an entry is to be set to active or inactive.
Input	debug	Boolean [Y N] parameter to provide verbose mode information. The default value is: N.

Returns

- None

Calls

- [m_log_set_options.sas](#)
- [m_utl_dec_passwd.sas](#)
- [m_utl_print_message.sas](#)
- [m_utl_print_mtrace.sas](#)
- [m_utl_unique_number.sas](#)

Usage

Example 1: Show help information:

```
%m_sys_upd_dbaccess(?)
```

Example 2 - Step 1: Create new DBACCESS table (Mode=C):

```
%m_adm_ctrl_dbaccess(  
    dstable = WORK.dbaccess  
    , dskey  = aespaskey  
    , mode   = C  
);
```

Example 2 - Step 2: Insert a new Database entry:

```
%m_adm_ctrl_dbaccess(  
    dstable = WORK.dbaccess  
    , dskey  = aespaskey  
    , dnam   = TEMP  
    , dtyp   = SAS  
    , dpth   = %str(%sysfunc(getoption(WORK)))  
    , dacc   = T  
    , mode   = I  
);
```

Example 2 - Step 3: Deactivate Database entry:

```
%m_sys_upd_dbaccess(  
    dstable = WORK.dbaccess  
    , dskey  = aespaskey  
    , dname  = TEMP  
    , daccs  = T  
    , ddact  = N  
);
```

Example 2 - Step 4: View DBACCESS entry status:

```
%m_adm_ctrl_dbaccess(  
    dstable = WORK.dbaccess  
    , dskey  = aespaskey  
    , dsel   = %str(dbname='TEMP')  
    , mode   = V  
);
```

Example 2 - Step 5: Reactivate Database entry:

```
%m_sys_upd_dbaccess(  
    dstable = WORK.dbaccess  
    , dskey  = aespaskey  
    , dname  = TEMP  
    , daccs  = T  
    , ddact  = Y  
);
```

Copyright

Copyright 2008-2020 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/>>.