rxcmanager_eng_guide

COLLABORATORS

	TITLE : rxcmanager_eng_guide		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

rxcmanager_eng_guide

1.1 Index

```
                    rxcmanager.library 5.5

    1.
                Warning
                    2.
                About this guide
                    2.
                Distribution
                    3.
                Author
                    4.
                Introduction
                    5.
                System requirements
                    6.
                Installation
                    7.
                Terms
                    8.
                Bugs
                    9.
                Functions
```

1.2 warning

```
    THIS SOFTWARE AND INFORMATION ARE PROVIDED "AS IS".
    ALL USE IS AT YOUR OWN RISK, AND NO LIABILITY OR
    RESPONSIBILITY IS ASSUMED. NO WARRANTIES ARE MADE.
```

1.3 guide

This guide contains links with inline ARexx macros wich use rmh.library/OpenURL. They will work if and only if you have both openurl.library and rmh.ibrary installed.

rmh.library is by me and can be found on aminet at rexx/utils/rmh.lha

openURL.library is by Troels Walsted Hansen

"This library was created to make it easier for application programmers to include clickable URLs in their applications, about windows, etc. ..."

You can find it on aminet at comm/www/OpenURL20.lha

1.4 distribution

rxcmanager.library is FreeWare.

You are free to detribute it as long as the original archive is kept intact. Commercial use or its inclusion in other software package is prohibited without prior consens from the Author.

1.5 author

I am Alfonso Ranieri .

My e-mail address is alfier@iol.it .

My home page is at http://users.iol.it/alfier/ .

You can find me on:
- #amigaita ircnet ;
- #amyita   ircnet .

1.6 introduction

The library is a basic CManager databases query functions set to use in ARexx macros.

CManager is a great resource handler by Simone Tellini.

1.7 requirements

This library needs AmigaOS, version >=2 and, of course, CManager .

1.8 installation

Use the installation script or just Click here .

1.9 terms

- stem or stemName: a valid ARexx variable name e.g. var, var.0, var.name, var ;

- types of arguments: the types used are:

D	any data	--
N	numeric	/N
S	symbol	/S ARexx valid symbol
V	stemName	/V ARexx valid symbol as S but with length<20

1.10 bugs

None known.

1.11 functions

CMAddEntry

CMClose

CMFind

CMLoad

CMUserData

CMFindVirtualField

CMGetEntry

CMStart

CMUserData

help

1.12 cmaddentry

```
                CMAddEntry

Usage: res = CMAddEntry(type,stem)
<type/D>,<stem/V>

Add a CM entry in the "Genesis global" user data.

"type" defines the type of the entry to add:
- USER
- WWW
- FTP
- CHAT

The entry is read from the "set" fields of "stem";
see
             Fields
              for a list of the "set" fields.

Return a boolean, result of the operation.
```

1.13 cmclose

```
CMClose

Usage: call CMClose()
-

Close an istance of CM created with CMStart() .
```

1.14 cmfind

```
                CMFind

Usage: num = CMFind(where,store,keys,askUser)
<where>,<store/V>,[key/V],[askUser/N]

Find a CM entry in "where" group(s) by AmigaDOS patterns defined in "keys"
and write the results in "store".

"where" can be one or more of
- USER
- URL
- FTP
- CHAT
- ALL    all the above
- GROUPS if specified, empty groups are are returned.
and the search is made in the group(s) choosen.

The entries are written in "stem.0", ..., "stem.j" where j is
the result of the function minus 1.
```

You can set the max number of entries returned, setting in "store" the field
- MAX

to a positive number.
If MAX is set to x than only x entries are returned.

The searching is done in this order:
- USER
- WWW
- FTP
- CHAT

(of course only for the types you choosed).

"keys" is a stem whose fields are to be set to the pattern, function should search for. It is optional and no key or no fields defined means get all entry.

"askUser" is an ARexx boolean:
- 0 don't ask user to login (default);
- 1 ask user to login.

See Fields for a list of the "searching" fields and "results" fields.

Returns the number of the entries found:
- if GROUPS was specified, 1 means none;
- if GROUPS was NOT specified, 0 means none;

1.15 cmfindvirtualfield

CMFindVirtualField

Usage: res = CMFindVirtualField(comment,field)
<comment>,<field>

I asked Simone to add a MASK field to USER and he answered "do that by yourself via COMMENT"... so that is.

Search in "comment" a **LINE** beginning with "field" and return an empty line on failure or the data after "field" (and spaces if any) till EOL.

This let you define *virtual fields*.

E.g. no birthday field in USER, so just add in COMMENT the line
#Birthday MM/DD/YY

After you get a user entry, lets say u.i, just search for a birthday via birthday = CMFindVirtualField(u.i.comment,"#Birthday")

The searching is no case.
Must be 2 <= length(field) <= 32

1.16 cmgetentry

```
                 CMGetEntry

Usage: res = CMGetEntry(what,stem)
<what>,<stem/V>

Gets CM entry from the current CM data.

You must open CM before calling this function with CMStart().

"type" defines what to get and if "multiselect"; it is one or more of
- USER
- WWW
- FTP
- CHAT
- MULTISELECT

If "MULTISELECT" is not set, datas will be written in "stem".
If multiselect is set, datas will be written in "stem.0",...,"stem.j"
where j is the number of the selected entries minus 1.

See
             Fields
              for a list of the "result" fields.

Returns the number of the selected entries, so 0 means none.
```

1.17 cmload

```
CMLoad

Usage: res = CMLoad(file,user)
<file>,[user]

Loads a data file for the current macro.
rxcmanager will use that data in the current macro.
"file" is a data file, e.g user/user.data.
"user" is an optional user name: data will be load if and only if data
belongs to that user.

NOTA BENE: CManager path can be found in ENV:Vapor/CManager_path .

Returns an arexx boolean.
```

1.18 cmuserdata

```
CMUserData

Usage: res = CMUserData(user)
<user>
```

Loads the "user" data file for the current macro. rxcmanager will use that data in the current macro, eccept for CMAddEntry() that always add the entry to the global user data.

Data will be load if and only if data belongs to that user.

Returns an arexx boolean.

1.19 cmstart

CMStart

```
Usage: res = CMStart(file,pubScreen,close)
[file],[pubScreen],[close/N]
```

Opens a CM istance with "file" as datas on the "pubScreen".
"close" is an ARexx boolean:
- 0 means don’t close this CM at macro exist (default);
- 1 close this CM at macro exit.

Returns an ARexx boolean.

1.20 help

help

```
Usage: pattern = help(funName)
<funName>
```

Returns the arguments pattern of "funName".

1.21 fields

The "searching" fields are the fields user can defined in "key" arguments of the function CMFind. The value of the fields is an AmigaDOS pattern of max 50 chars (after parsing: it means about 20 chars before).

They are:

- USER
  . NAME
  . ADDRESS
  . CITY
  . COUNTRY
  . ZIP
  . COMMENT
  . ALIAS
  . PHONE
  . FAX
  . EMAIL

```
  . WWW
  . FTP
  . LASTNAME
  . MOBILE
  . ICQ
  . COMPANY
  . BADDRESS
  . BCITY
  . BCOUNTRY
  . BZIP
  . BPHONE
  . BFAX
  . BEMAIL
  . BJOBTITLE
  . BDEPARTMENT
  . BOFFICE

- WWW
  . NAME
  . WWW
  . COMMENT
  . WEBMASTER
  . EMAIL

- CHAT
  . CHANNEL
  . SERVER
  . MAINTAINER
  . NICK
  . WWW
  . COMMENT
  . PASSWORD

- FTP
  . NAME
  . FTP
  . COMMENT
  . USERNAME
  . PASSWORD
  . LOCAL
```

The "result" and the "set" fields are the fields of "stem" written by CMGetEntry and CMFind or read by CMAddEntry.

They are:

```
- USER
  . TYPE        "USER"
  . GROUP       name of the group (empty if in main group). Only for CMFind() .
  . PARENTS     parents list separated by '/' . See note below.
  . EMPTYGROUP  See note below.
  . NAME
  . ADDRESS
  . CITY
  . COUNTRY
  . ZIP
```

```
      . COMMENT
      . ALIAS
      . PHONE
      . FAX
      . EMAIL
      . WWW
      . FTP
      . LASTNAME
      . MOBILE
      . ICQ
      . COMPANY
      . BADDRESS
      . BCITY
      . BCOUNTRY
      . BZIP
      . BPHONE
      . BFAX
      . BEMAIL
      . BJOBTITLE
      . BDEPARTMENT
      . BOFFICE

    - WWW
      . TYPE        "WWW"
      . GROUP       name of the group (empty if in main group). Only for CMFind() .
      . PARENTS     parents list separated by '/' . See note below.
      . EMPTYGROUP  See note below.
      . NAME
      . WWW
      . COMMENT
      . WEBMASTER
      . EMAIL

    - CHAT
      . TYPE        "CHAT"
      . GROUP       name of the group (empty if in main group). Only for CMFind() .
      . PARENTS     parents list separated by '/' . See note below.
      . EMPTYGROUP  See note below.
      . CHANNEL
      . SERVER
      . MAINTAINER
      . NICK
      . WWW
      . COMMENT
      . PASSWORD

    - FTP
      . TYPE        "FTP"
      . GROUP       name of the group (empty if in main group). Only for CMFind() .
      . PARENTS     parents list separated by '/' . See note below.
      . EMPTYGROUP  See note below.
      . NAME
      . FTP
      . COMMENT
      . USERNAME
      . PASSWORD
      . PORT
```

. RETRIES
. LOCAL

Note:

The field PARENTS is valid if and only if the parents string is less than 512 chars. If groups are so nested that their names list is longer than 512 chars the result is undefined .

The field EMPTYGROUP is a boolean, meaningfull iff in CMSearch was specified GROUPS; it is

- 0 if the group is not empty
- 1 if the group (of name .GROUP) is empty.

It is needed in export macros.