cxobj_database

COLLABORATORS

	TITLE : cxobj_database	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 cxobj_database **1**

1.1 AMIGA-E Modules: cxobj_oo/Main . . . 1
1.2 AMIGA-E Modules: cxobj/cxobj() . . . 2
1.3 AMIGA-E Modules: cxobj/create() . . . 2
1.4 AMIGA-E Modules: cxobj/error() . . . 2
1.5 AMIGA-E Modules: cxobj/get() . . . 3
1.6 AMIGA-E Modules: cxobj/getport() . . . 3
1.7 AMIGA-E Modules: cxobj/signal() . . . 3
1.8 Author's Infos . . . 3
1.9 AMIGA-E Modules: cxobj/EXAMPLE PROGRAM . . . 4
1.10 ERROR TABLE . . . 5
1.11 AMIGA-E Modules: rxobj_oo/Introduction . . . 5

Chapter 1

cxobj_database

1.1 AMIGA-E Modules: cxobj_oo/Main

```
                                   ** CXOBJ_OO - Written By Fabio Rotondo **

                              ** DOCUMENTATION GUIDE **


                    Introduction

                   Author's Infos

                   Example Program
                      COMMANDS                                BRIEF DESCRIPTION
   ------------------------------------------------------------------------------

                   cxobj()
                   Initializes the cxobj

                   create(name, title, descr, pri)
                   Creates a Commodity

                   error()
                   Gets the last error

                   get()
                   Gets the next incoming message

                   getport()
                   Gets the Broker's MsgPort

                   signal()
                   Gets the Commo's SigBit



                   ERROR TABLE
                   ERROR TABLE DESCRIPTION
```

1.2 AMIGA-E Modules: cxobj/cxobj()

```
                      NAME: cxobj()

DESCRIPTION: This command initializes an cxobj.

     INPUTS: NONE.

    RESULTS: This command may fail.

   SEE ALSO:
               error()
```

1.3 AMIGA-E Modules: cxobj/create()

```
                      NAME: create(name:PTR TO CHAR, title, PTR TO CHAR, descr: ↩
                         PTR TO CHAR,
                    pri=0)

DESCRIPTION: this command creates a Commodity Object and links it to the
             Exchange List.

     INPUTS: name   - Name of the Commodity. It is the text which will appear
                      on the left of Exchange Window.

             title  - Title of the Commodity. It is first text line on the
                      right of Exchange Window.

             descr  - Description of the Commodity. It is the second line of
                      text appearing on the right side of Exchange Window.

             pri    - (optional) Commodity Priority.

    RESULTS: TRUE   - Everything was fine.
             FALSE  - An error occurred. Check
                error()
                   SEE ALSO:
                error()
```

1.4 AMIGA-E Modules: cxobj/error()

```
                      NAME: error()

DESCRIPTION: this command returns the LAST error occurred.

     INPUTS: NONE.

    RESULTS: 0 = No errors.

   SEE ALSO:
```

```
                ERROR TABLE
```

1.5 AMIGA-E Modules: cxobj/get()

```
        NAME: get()

 DESCRIPTION: this command gets the next waiting incoming Exchange Message.

      INPUTS: NONE.

     RESULTS: a MsgID    in case it was an CXM_COMMAND
              CXM_IEVENT in case it was a CXM_IEVENT
              FALSE      in case nothing was waiting.

    SEE ALSO:
```

1.6 AMIGA-E Modules: cxobj/getport()

```
        NAME: getport()

 DESCRIPTION: This command returns the broker's MessagePort.

      INPUTS: NONE.

     RESULTS: PTR TO Broker's MsgPort.

    SEE ALSO:  create()
```

1.7 AMIGA-E Modules: cxobj/signal()

```
                      NAME: signal()

 DESCRIPTION: This command returns the SigBit of the cxobj.

      INPUTS: NONE.

     RESULTS: a LONG var containing the SigBit.

    SEE ALSO:
              create()
```

1.8 Author's Infos

```
My name is Fabio Rotondo. I am a free-lance Amiga programmer and I would
like to get in touch with anyone who writes code for the Amiga.
I write in AmigaE, BlitzII, C and a bunch of other languages.
```

```
    Please, feel free to contact me for any suggestions/questions.

    My address is:

    Fabio Rotondo
    C.so Vercelli 9
    28100 Novara
    ITALY
    Tel. (ITA) - (0)321 - 459676
    e-mail: fabio.rotondo@intercom.it

    Thanks!
```

1.9 AMIGA-E Modules: cxobj/EXAMPLE PROGRAM

```
MODULE 'libraries/commodities','Fabio/cxobj_oo'  -> Needful things

PROC main()
  DEF myval, txt
  DEF commo=NIL:PTR TO cxobj        -> Our cxobj!!!

  NEW commo.cxobj()                 -> Let's initialize it!

  IF commo                          -> Have We Succeded ???
    IF commo.create('FabioCx','Fabio Commodity', 'The Commodity') -> Create!
      WriteF('Commodity Running...\n')   -> Signal the user we are running...
      REPEAT
        Wait(commo.signal())             -> Waiting msgs from Exchange
        IF (myval:=commo.get())          -> Gets the msgs
          SELECT myval                   -> What kind of msg arrived??
            CASE CXM_IEVENT
              txt:='IEVENT'
            CASE CXCMD_DISABLE
              txt:='DISABLE'
            CASE CXCMD_ENABLE
              txt:='ENABLE'
            CASE CXCMD_KILL
              txt:='KILL'
            CASE CXCMD_UNIQUE
              txt:='Unique'
            CASE CXCMD_APPEAR
              txt:='Appear'
            CASE CXCMD_DISAPPEAR
              txt:='Disappear'
          ENDSELECT
          WriteF('Msg: \s\n', txt)      -> Describe it to the user.
        ELSE
          WriteF('get() error:\d\n', commo.error())  -> Ouch! An error!
        ENDIF
      UNTIL myval=CXCMD_KILL            -> The user pressed "KILL"
      WriteF('Exiting...\n')            -> We are gonna DIE!!!
      END commo                         -> ALWAYS _END_ THE CXOBJ!!!
    ELSE
      WriteF('Broker Error:\d\n', commo.error()) -> Ouch! Broker's Error!
```

```
    ENDIF
  ELSE
    WriteF('Init Error:\d\n', commo.error()) -> Ouch! Init Erorr!
  ENDIF

  CleanUp(0)                              -> Let's Keep Things Clean.
ENDPROC
```

1.10 ERROR TABLE

VALUE	NAME	DESCRIPTION
1	CXLIB_FAIL	Failed To Open 'commodities.library'.
2	MSGPORT_ERROR	Failed to CreateMsgPort().
3	BROKER_ERROR	Failet to CxBroker() the Commodity.

1.11 AMIGA-E Modules: rxobj_oo/Introduction

```
                    CXOBJ_OO - Introduction

    cxobj_oo is a little AmigaE "class" which helps the programmer to handle
    Commodities. With this little module, you can easyly transform your program
    into an Exchange Object (CXOBJ) without worrying about what there is
    "inside" the module.

    The cxobj_oo.m file should be placed inside a new dir you created called
    ** EMODULES:Fabio ** this will allows future compatibility with my new
    modules and also will save you from "losting" my module inside the mess
    of all the others... like I did so many times before creating this holy
    directory!!!

    The use is quite simple. Just DEFine a PTR TO cxobj at the beginning of
    your program and initialize it with a NEW objname.cxobj().
    All the rest is funny! Look at the
                Example Program
                for having
    an idea.

    cxobj_oo has been developed and is (C)Copyright by Fabio Rotondo.

    SEE ALSO:
                Author's Infos
```