

**rxobj\_database**

**COLLABORATORS**

	<i>TITLE :</i> rxobj_database		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 16, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>rxobj_database</b>	<b>1</b>
1.1	AMIGA-E Modules: rxobj_oo/Main	1
1.2	Author's Infos	2
1.3	AMIGA-E Modules: Example Program	2
1.4	AMIGA-E Modules: rxobj_oo/Introduction	3
1.5	AMIGA-E Modules: rxobj_oo/rxobj()	3
1.6	AMIGA-E Modules: rxobj_oo/error()	4
1.7	AMIGA-E Modules: rxobj_oo/get()	4
1.8	AMIGA-E Modules: rxobj_oo/reply()	4
1.9	AMIGA-E Modules: rxobj_oo/signal()	5
1.10	AMIGA-E Modules: rxobj_oo/send()	5
1.11	AMIGA-E Modules: rxobj/ERROR TABLE	5
1.12	AMIGA-E Modules: rxobj_oo/getanswer()	6
1.13	AMIGA-E Modules: rxobj_oo/createPort()	6
1.14	AMIGA-E Modules: rxobj_oo/deletePort()	7
1.15	AMIGA-E Modules: rxobj_oo/result()	7
1.16	AMIGA-E Modules: rxobj_oo/rc()	7

## Chapter 1

# rxobj\_database

### 1.1 AMIGA-E Modules: rxobj\_oo/Main

\*\* RXOBJ\_00 - Written By Fabio Rotondo \*\*

\*\* DOCUMENTATION GUIDE \*\*

Introduction

Author's Infos

Example Program

COMMANDS

BRIEF DESCRIPTION

---

rxobj(portname)  
Initializes the object

createPort(pname, pri)  
Creates a Message Port

deletePort(port)  
Deletes a Message Port

error()  
Gets the last error

get(parseproc)  
Gets the next incoming ARexx Message

getanswer()  
Waits for replies by another Rexx Host

rc()  
Returns the last RC got

reply(rxmsg, rc=0, res=NIL)  
Reply to an ARexx Message

---

```
result()
Return the last RESULT string got

send(hname, comm, unkmsg, flags)
Sends a message to an ARexx Host

signal()
Returns the SigBit

ERROR TABLE
ERROR TABLE DESCRIPTION
```

## 1.2 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: fsoft@intercom.it
WWW: http://www.intercom.it/homepages/utenti/fsoft/index.html
```

Thanks!

## 1.3 AMIGA-E Modules: Example Program

```
MODULE 'Fabio/rxobj_oo'      -> The MAGIC module should be in: EMODULES:Fabio dir

DEF quit = FALSE

PROC main()
  DEF rx:PTR TO rxobj      -> Our ARexx object
  DEF rxsig
  DEF sig

  NEW rx.rxobj('FabioSoft') -> Here we init the obj with a Portname

  WriteF('Host: "FabioSoft" Alive AND Kicking!\n')
  WriteF('Send a QUIT message TO the port!\n')

  rxsig:=rx.signal()      -> This is our port SigBit
  REPEAT
```

```

    sig:=Wait(rxsig)           -> Waiting for Arexx commands
    rx.get({parse})           -> If any, parse them
UNTIL quit=TRUE

WriteF('This is the end...\n')
END rx
ENDPROC

PROC parse(t)
WriteF('Parsing:\s\n', t)    -> Variable t is a string...
IF StrCmp(t, 'QUIT')
    quit:= TRUE
ENDIF
ENDPROC TRUE, 0,NIL

```

## 1.4 AMIGA-E Modules: rxobj\_oo/Introduction

### RXOBJ\_OO - Introduction

rxobj\_oo is a little AmigaE "class" which helps the programmer to handle ARExx Ports. With this little module, you can easily add an ARExx port to your programs and use it without worrying about what there is "inside".

The rxobj\_oo.m file should be placed inside a new dir you created called **\*\* EMODULES:Fabio \*\*** this will allow future compatibility with my new modules and also will save you from "losing" 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 rxobj at the beginning of your program and initialize it with a NEW objname.rxobj(portname).

All the rest is funny! Look at the

Example Program

for having

an idea.

rxobj\_oo has been developed and is (C)Copyright by Fabio Rotondo.

SEE ALSO:

Author's Infos

## 1.5 AMIGA-E Modules: rxobj\_oo/rxobj()

NAME: rxobj(portname:PTR TO CHAR)

DESCRIPTION: This command initializes an rxobj and opens the Message Port portname.

INPUT: portname - Name of the MsgPort.

RESULTS: NONE.

SEE ALSO:  
error()

## 1.6 AMIGA-E Modules: rxobj\_oo/error()

NAME: err := error()

DESCRIPTION: returns the last generated error.

INPUT: NONE.

RESULTS: err - a LONG containing the last error.

SEE ALSO:  
ERROR TABLE

## 1.7 AMIGA-E Modules: rxobj\_oo/get()

NAME: get(parseproc)

DESCRIPTION: gets the next ARexx message waiting and calls the parsing PROC to handle it.

INPUT: parseproc - This is the address of the parsing PROC.  
The parsing proc MUST do these things:

- \* get a PTR TO CHAR as an argument.
- \* return three values in this order:
  - isrxmessage (TRUE/FALSE - TRUE = recognised rexx command)
  - rc = return code
  - result = result string

NOTE: The get() command automagically calls the reply() method...

RESULTS: NONE.

SEE ALSO:  
reply()

## 1.8 AMIGA-E Modules: rxobj\_oo/reply()

---

NAME: reply(rexxmsg:PTR TO rexxmsg, rc, resultstr)

DESCRIPTION: reply to an ARexx Message, setting RC and RESULT string.

INPUT: rexxmsg           - PTR TO rexxmsg, it is the message we want  
                          to reply() to.  
      rc                 - RC of the ARexx Function.  
      resultstr         - RESULT ARexx String.

RESULTS: NONE.

SEE ALSO:  
      get()

## 1.9 AMIGA-E Modules: rxobj\_oo/signal()

NAME: signal()

DESCRIPTION: returns the value of the sigbit mask of rxobj.

INPUT: NONE.

RESULTS: a LONG value containing the sigbit.

SEE ALSO:  
      rxobj()

## 1.10 AMIGA-E Modules: rxobj\_oo/send()

\*\*\* NOTE: THIS COMMAND IS STILL EXPERIMENTAL !!! \*\*\*

NAME: send(hostname, command, unknownmsg, flags)

DESCRIPTION: sends a message to an ARexx Host.

INPUT: hostname   - name of the ARexx Port of the Host.  
      command     - command to send to the ARexx Host.  
      unknownmsg  - original msgpointer (usually NIL)  
      flags       - usually 0

RESULTS: returns a rexxmsg for a succeeded operation,  
          0 if it will fail.

SEE ALSO:

## 1.11 AMIGA-E Modules: rxobj/ERROR TABLE

---



\*\* NOTE: These error may "grow" in future!!!

VALUE	NAME	MEANING
1	DOUBLE_PORT	The programmer tried to open a MsgPort twice with the same Name.
2	SIGNAL_ALLOCATION	The rxobj failed creating the SigBit.

well, at the moment these are the only two errors the module can handle with. In future, I'll promise you more and more accurate checks.

## 1.12 AMIGA-E Modules: rxobj\_oo/getanswer()

NAME: getanswer()

DESCRIPTION: This command waits for an answer by another Rexx Host. This is usefull if you want to get the RC or Result String from an Host where you have just sent a message.

INPUT: NONE.

RESULTS: NONE.

NOTE: This command is DANGEROUS: you will lost any command sent to your Host by other sites.

SEE ALSO:

result()

rc()

## 1.13 AMIGA-E Modules: rxobj\_oo/createPort()

NAME: createPort(portname, priority)

DESCRIPTION: This command creates a Message Port. It is used by rxobj() itself, but it may turn to be usefull also for other purposes.

INPUT: portname - Name of the port, if it is NIL, the port will be PRIVATE.

priority - Public Port Priority.

RESULTS: NONE.

NOTE: Trying to create an EXISTING public Message Port will not get to any error() message.

SEE ALSO:

deletePort

## 1.14 AMIGA-E Modules: rxobj\_oo/deletePort()

NAME: deletePort(port:PTR TO mp)

DESCRIPTION: This command deletes a Message Port.  
It is used by rxobj() itself, but it may turn to be usefull  
also for other purposes.

INPUT: port - PTR TO a message port structure.

RESULTS: NONE.

NOTE: Trying to delete a unexistnsing Message Port will not get  
to any error() message.

SEE ALSO:

createPort

## 1.15 AMIGA-E Modules: rxobj\_oo/result()

NAME: result()

DESCRIPTION: This command will return the last result obtained by a  
getanswer() or get() method.

INPUT: NONE.

RESULTS: a PTR TO CHAR containing the result string.

SEE ALSO:

rc()

get()

getanswer()

## 1.16 AMIGA-E Modules: rxobj\_oo/rc()

NAME: rc()

DESCRIPTION: This command will return the last rc code obtained by a  
getanswer() or get() method.

INPUT: NONE.

RESULTS: a LONG containing the rc code.

---

SEE ALSO:

`result()`

`get()`

`getanswer()`

---