

# **ARexxSupervisor**

**COLLABORATORS**

	<i>TITLE :</i> ARexxSupervisor		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 27, 2024	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ARexxSupervisor</b>	<b>1</b>
1.1	ARexxSupervisor.doc - Table of contents	1
1.2	Copyright,distribution and disclaimer	1
1.3	Credits and special thanks	2
1.4	System Requirements	2
1.5	What is ARexxSupervisor ?	3
1.6	Starting ARexxSupervisor	3
1.7	Starting ARexxSupervisor from CLI	3
1.8	Starting ARexxSupervisor from Workbench	3
1.9	ARexxSupervisor's window	3
1.10	? gadget / ? key: Information	4
1.11	Help gadget / Help key: Show help file	4
1.12	I gadget: Iconification	4
1.13	Q gadget / Q key / close gadget: Quit program	5
1.14	Halt gadget / H key: Stop execution of all ARexx programs	5
1.15	Global trace gadget / G key: force all ARexx programs into trace mode	5
1.16	Port gadget / P key: destination port name	5
1.17	Type gadget / Y key: type of the RexxMsg	6
1.18	Send command	6
1.19	Invoke function	6
1.20	Close tracing console	7
1.21	Open tracing console	7
1.22	Add clip entry	7
1.23	Remove clip entry	7
1.24	Add library	7
1.25	Remove library	8
1.26	FileExt gadget / X key: File extension	8
1.27	StdIn gadget / I key: Standard input stream	8
1.28	StdOut gadget / O key: Standard output stream	8
1.29	ARGx gadgets / 0..F keys: Argument strings	8

---

1.30 NoIO gadget / N key: NoIO modifier flag . . . . .	8
1.31 Result gadget / R key: Result modifier flag . . . . .	9
1.32 String gadget / S key: String modifier flag . . . . .	9
1.33 Token gadget / T key: Token modifier flag . . . . .	9
1.34 Send message gadget/ M key: Send message to host . . . . .	9
1.35 Result1 gadget: Primary result code . . . . .	9
1.36 Result2 gadget: Secondary result code . . . . .	10
1.37 Translation . . . . .	10
1.38 Future improvements . . . . .	10
1.39 Contacting the author . . . . .	10

---

## Chapter 1

# ARexxSupervisor

### 1.1 ARexxSupervisor.doc - Table of contents

Welcome to ARexxSupervisor 1.10, by Frédéric DELACROIX. This is the doc file that should always be with the program for public distribution. And now, like every reasonable guide file, here is a TOC:

#### TABLE OF CONTENTS

COPYRIGHT, DISTRIBUTION and DISCLAMER <-- This is very important!  
CREDITS and SPECIAL THANKS

SYSTEM REQUIREMENTS <-- My advise is you check this!  
WHAT IS AREXXSUPERVISOR ?  
STARTING AREXXSUPERVISOR  
AREXXSUPERVISOR'S WINDOW

FUTURE IMPROVEMENTS  
TRANSLATION  
CONTACTING THE AUTHOR <-- Don't forget !

### 1.2 Copyright, distribution and disclaimer

ARexxSupervisor 1.10 was entirely written by F.DELACROIX on the wonderful HiSoft DevPac 3. IT REMAINS COPYRIGHTED BY ME. However, the program is released as SHAREWARE. This means that public distribution is allowed (and encouraged) as long as:

- NO profit is made beyond MINIMAL copy&posting fees (you may not charge more than Fred Fish does for a single AmigaLib Disk)

- The WHOLE archive (or, say, all files) is (are) distributed UNMODIFIED IN ANY WAY. There is although one exception: as ARexxSupervisor is provided in two languages (for now), french and english, you may spread only one of them. Yet, I'd prefer you to distribute both if you are releasing ARexxSupervisor on a wide-spread international PD-library.

Please note also that ARexxSupervisor is SHAREWARE, meaning that if you continue use it after an evaluation period you MUST send me a contribution (suggested amount is \$10 or equivalent). The source is available from me, just add \$10 dollars more. My address can be found at the end of this document.

As always, I must mention that I do not make ANY guarantee that the program, although tested a lot, does not contain any bug, and, therefore, should a problem arise and cause direct or indirect damage, I CANNOT BE HELD RESPONSIBLE FOR ANYTHING. YOU USE AREXXSUPERVISOR AT YOUR OWN RISK.

### 1.3 Credits and special thanks

ARexxSupervisor was written on an Amiga 500 (OS 2.04), running the marvelous HiSoft Devpac 3.

ARexx is copyright © 1987 by William S. Hawes.

The great reqtools.library is Copyright © Nico François (it is not supplied in this archive but you could find it everywhere in public domain disk libraries).

The Graphic User Interface was created by GadToolsBox 2.0 (available on Fish 852) which is ©1991-93 Jaba Development, optimized and adapted by myself.

As of V1.10, ARexxSupervisor makes use of the errormsg.library which is copyrighted by me. You can find it in PD collections.

My special thanx and greetinks to:

- Serge HAMMOUCHE from France-Festival-Distribution
- Commodore and Jay Miner for designing the Amiga ;-)
- Yves QUIQUEMPOIS
- François JACOBS
- Philippe DUSSART

### 1.4 System Requirements

ARexxSupervisor will need the following to run:

- Kickstart V37 or newer (OS 2.04), it will not run on the old OS 1.3. The only thing you can do in this case is... UPGRADE !

- reqtools.library V38+ © Nico François, which can be found nearly everywhere (not Nico, the library :-)

- errormsg.library V1.0+, ©F.Delacroix, check out PD libraries.

- ARexx. This is normally provided on your Workbench 2.0 disk.

Note that you must run REXXMast BEFORE running ARexxSupervisor (this is normally done is the 2.0 startup-sequence)

---

- the amigaguide.library (which you possess, because you are reading this file). It can be found on Fish 870. However it is not a fatal error not having installed this library, the online-help feature will simply be disabled.

- some knowledge about programming & debugging ARexx interfaces!

## 1.5 What is ARexxSupervisor ?

ARexxSupervisor is mainly a tool for the developers of you who want to integrate and properly test/debug ARexx interfaces in their programs.

ARexxSupervisor provides a means of mouse-controlling the data that is sent to a valid ARexx port. Be careful however, it is easy to make destructive actions or provoke deadlocks or gurus.

## 1.6 Starting ARexxSupervisor

ARexxSupervisor can be launched either from CLI or from Workbench.

## 1.7 Starting ARexxSupervisor from CLI

CLI usage is: ARexxSupervisor [PUBSCREEN <name>]  
PUBSCREEN/K

This means that you can make ARexxSupervisor's window appear on any public screen that is already opened. Example:

ARexxSupervisor PUBSCREEN DEVPAC.1 or ARexxSupervisor PUBSCREEN=DEVPAC.1 will make ARexxSupervisor use Devpac 3's screen. The default is to use the default public screen (normally the Workbench screen). If the public screen is not opened or is in private state, ARexxSupervisor "falls back" to the default.

## 1.8 Starting ARexxSupervisor from Workbench

You start ARexxSupervisor from the Workbench by simply double-clicking its icon. There is currently one tooltype defined: PUBSCREEN=<name>. See CLI usage for more information.

## 1.9 ARexxSupervisor's window

When ARexxSupervisor comes up, it opens a large window (it should however fit on an NTSC screen), with a nice GadTools interface made of a bunch of gadgets. Those of you who are already familiar with ARexx will recognize all fields of an ARexxMessage structure, plus some more auxiliary gadgets. Let's see their use individually:

<u>?</u>	Type	Result
Help	<u>FileExt</u>	String
<u>I</u>	StdIn	Token
<u>Q</u>	StdOut	Send Message
Halt	<u>ARGx</u>	(x=0...F) Result1
Global Trace	NoIO	Result2
Port		

Please note that most of these gadgets have an underlined character, meaning that their functions can be accessed by the keyboard by pressing the given key. Exceptions to this rule are the iconification gadget (which has no keyboard shortcut), and the Help gadget, which can be accessed via the Help key. Of course, for these shortcuts to work, there must not be an active string gadget.

Another feature is the Help key, which, when pressed while a string gadget is active (apart from the Port and FileExt string gadgets) brings up a file requester. The contents of this file requester are then put in the gadget.

## 1.10 ? gadget / ? key: Information

Nothing really alarming: this brings up the information requester, showing some important things like the shareware notice and my address.

## 1.11 Help gadget / Help key: Show help file

This gadget displays this file (but maybe that's how you had it displayed just now!). For this feature to work correctly, you need amigaguide.library to be installed in your LIBS: directory (see SYSTEM REQUIREMENTS), otherwise it won't show up. The file you're reading now must be named ARexxSupervisor.guide and be in your amigaguide path (which can be set via the SetEnv command, see your amigaguide documentation). I suggest you copy it to your HELP: directory.

If the file is successfully displayed by AmigaGuide, the program is put to sleep until all AmigaGuide windows are closed.

## 1.12 I gadget: Iconification

This gadget is used to iconify the program when you think it takes too much room on your screen. You will of course need the workbench.library installed to do this. The result will be closing of the main window and a new AppIcon will be added on the workbench screen. If an error in creating the AppIcon occurs, the window won't close.

When you want the program to come back, just double-click on the AppIcon and a requester will pop up asking if you want to wake the program, quit or cancel the operation. Click on Wake up and the window will re-open. If it cannot (because of low memory...), you will have to wait some more. If you

chose Quit, the program will finish, unless it is still waiting for ARexx to reply a previously sent message (don't want to wake the guru, do you? :-).

Plus, if ARexxSupervisor was waiting for a reply, it will automatically wake up on receiving it. This is it.

### **1.13 Q gadget / Q key / close gadget: Quit program**

Do I really have to explain this function? This will cause ARexxSupervisor to end, unless it is waiting for a reply, in which case a requester would pop up, giving you the opportunity to iconify instead.

### **1.14 Halt gadget / H key: Stop execution of all ARexx programs**

This function is absolutely the same as if you typed HI in a Shell: all ARexx scripts will stop (unless they trapped this case) and will issue an "execution halted" message.

Note that this must be your LAST chance to stop a program that has gone out of control. Indeed, stopping all programs is not a polite thing to do; you could destroy important data for other processes.

### **1.15 Global trace gadget / G key: force all ARexx programs into trace mode**

This is a checkbox, so it has two states: activated or not. When it is not selected, all ARexx programs run normally. When it is checked, all ARexx programs will come into interactive trace mode. For the execution of a program to continue, you must press RETURN in its output console (it MUST be an interactive console) or in the global tracing console if it is opened. So you can examine variables... to see what is wrong with your program.

But again, beware that is function affects the whole environment, and thus must be used with caution.

Note: you would get the same result by running the program TS for set and TE for clear.

### **1.16 Port gadget / P key: destination port name**

Enter there the name of the port to be addressed for sending the message. Note that this name IS case-sensitive. The port need not be opened at that time, but it will have to when you "Send Message". This will for example be AMIGAGUIDE.1 for the first occurrence of an amigaguide process. Also, be careful what you send to someone. Normally, for example, only ARexx is able to understand messages like Open Tracing Console. It is very easy to send nonsense messages!

---

The port at which you can reach ARexx's master process is named REXX.

## 1.17 Type gadget / Y key: type of the RexxMsg

This cycle gadget controls the type of the Message that is to be sent.

These types are:

Send command	Add clip entry
Invoke function	Remove clip entry
Close tracing console	Add library
Open tracing console	Remove library

At send time, the type field will be or'ed with the modifier flags (only significant ones). Normally, only Send command and Invoke function should be sent to other hosts than REXX.

## 1.18 Send command

This means that the message is a command to be executed by the host. ARG0 contains normally the name of the command (which is normally a file name for REXX) followed by the argument string. ARG1 to ARGF are unused.

The "Token" modifier flag however forces ARexx to tokenize the whole string, which can then have many argument strings.

"NoIO" will prevent the new process from inheriting the StdIO streams.

The "String" modifier will tell REXX this is "string file" rather than a file name.

The "Result" modifier will request the host to send a result for this command (commands generally do not return a result).

## 1.19 Invoke function

The message will request the host to perform an action, from which it should return a result. The function name is in ARG0 and the arguments are in ARG1 to ARGF.

A result is requested by supplying the corresponding modifier flag. The "String" and "NoIO" modifiers have the same effect as for Send Command.

The result will be placed in Result2 if there was no error (if Error1 was 0).

---

## 1.20 Close tracing console

This message will make the global tracing console close if it was open.

It is only meaningful for the REXX port.

## 1.21 Open tracing console

This will make the global tracing console open, and all programs that are in trace mode will send their tracing information to this window instead of their StdErr or StdOut streams (which might not be interactive!). However, it can be confusing if several programs are in trace mode at a time.

Send only to REXX.

## 1.22 Add clip entry

ARexx maintains a "clip list" which is shared by all processes. This adds an entry to this list. ARG0 is the name of the clip entry, ARG1 is its value. ARG2 is automatically filled with the length of ARG1. This is one restriction of using ARexxSupervisor: you cannot put any kind of data in a clip entry, you must use a string. Hope this won't matter too much.

Send only to REXX.

## 1.23 Remove clip entry

This will simply remove an entry from the clip list. ARG0 is the name of the entry to remove.

Send only to REXX.

## 1.24 Add library

This type will add a function library to ARexx's internal list, so that programs can access their functions.

ARG0 is the name of the library (ex: amigaguide.library),

ARG1 is the priority within this list. It must be a number between -100 and 100. (ex: 0)

ARG2 is the query offset for the library. For amigaguide, this is -30.

ARG3 is the version requested for the library. 0 will match any version.

This message is intended for REXX only.

---

## 1.25 Remove library

This message will of course remove the specified library from ARexx's internal library list.

ARG0 is the name of the library to remove. ex:amigaguide.library.

## 1.26 FileExt gadget / X key: File extension

This is the file extension that will be appended to a filename when searching for an ARexx program. This mechanism is used to differentiate programs written for different hosts (.rexx,.editor, etc...).

## 1.27 StdIn gadget / I key: Standard input stream

This is the stream that will be used for input. If This is not a valid filename (like CON:////Hello), then the StdIn will be the same stream as StdOut. And If StdOut is not valid either, NIL: will be used.

Please remember that this stream is ALWAYS opened by ARexxSupervisor in MODE\_OLDFILE, whereas StdOut is ALWAYS opened in MODE\_NEWFILE. So be careful.

If you press help while the gadget is active,a file requester will appear.

## 1.28 StdOut gadget / O key: Standard output stream

Here is the other stream: the one that is used for outputs. Be careful that if you intend to use the tracing features of ARexx, you must provide an interactive stream,or deadlocks can happen. To be safe, you may first open the global tracing console. See description of the StdIn gadget for more info.

## 1.29 ARGx gadgets / 0..F keys: Argument strings

These 16 string gadgets are the argument that will be used for passing the message to the host. Their use can vary widely depending on the type of message, but the cases where all 16 slots are used are very scarce. In fact this can only happen for functions. If you press Help while one of these gadgets is active, a file requester appears.

## 1.30 NoIO gadget / N key: NoIO modifier flag

When checked,this modifier flag suppresses inheritance of the IO streams for the child process (StdIn and StdOut are ignored).

---

### 1.31 Result gadget / R key: Result modifier flag

This flag must be set when you want the command or function you invoke to return you a result.

### 1.32 String gadget / S key: String modifier flag

This modifier tells ARexx that the ARG0 string is not the name of a program but the program itself (it is a "String file").

### 1.33 Token gadget / T key: Token modifier flag

This flag tells ARexx to parse ARG0 so that the command receives not one but many argument strings.

### 1.34 Send message gadget/ M key: Send message to host

This button will actually create the REXXMSG structure and send it to the host whose name is given in the Port gadget. All fields are filled and the message is sent, ARexxSupervisor disables all its gadgets except iconification,help,information,Halt,Trace,Quit. Quitting is however disabled as long as the message is not replied. ARexx usually replies the message when the program is over. ARexxSupervisor then sets the Result gadgets accordingly, enables the gadgets again and resumes normal operation.

### 1.35 Result1 gadget: Primary result code

This is the primary result code from the function previously performed. Usually, a non-null value indicates an error code or severity level. If Result2 is empty, this is a return code from a function, shown as:  
Return code: %ld. If Result2 is not null, this an error severity level, shown as: Severity level: %ld , and Result2 contains the Error code:  
Error code: %ld.

If Result1 is NULL, Result2 may contain the result from the function or command that was requested by the Result modifier flag (still following?:-).

The value of this gadget may be found in the environment variable named AResult1.

---

### 1.36 Result2 gadget: Secondary result code

This is the secondary result from the previous call. For further information see the Result1 gadget.

The value of this gadget is copied into the environment variable named AResult2.

### 1.37 Translation

As of V1.10, ARexxSupervisor is totally localized, meaning that it can be made to run in your language provided that the good catalog file is available in your LOCALE:Catalogs/<lang> directory. For now, there is only a french catalog, so if you want a catalog for your language, you will have to do it yourself. The only thing you have to do is edit the file named ARexxSupervisor.ct and fill the blanks with your translations of the strings that are placed in comments. Then send me the resulting file so I can generate a catalog file and send it back to you. The catalog file will be included in the next release (with the proper credits, of course). Also, think of translating the guide file !

### 1.38 Future improvements

ARexxSupervisor is of course subject to improvements. If you can think of something you would like to see included in the next release, just let me know. Some of them I already have in mind:

- auto-RexxMast
- an ARexx port !
- whatever you can think of !

### 1.39 Contacting the author

Please don't forget that ARexxSupervisor is SHAREWARE. If you find it useful, send me a little contribution (I suggest \$10 or equivalent) to the address below.

Frédéric DELACROIX  
5 rue d'Artres  
59269 QUERENAING  
FRANCE, EARTH

Also, any bug report/suggestion/postcards is welcome.

---