

# **ARexx**

P!\K/AVOP|Un0X

COLLABORATORS

	TITLE : ARexx		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	P\K\AVOP\Un0X	June 8, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ARexx</b>	<b>1</b>
1.1	ARexx port . . . . .	1
1.2	ARexx - CloseEnvironment . . . . .	1
1.3	ARexx - Evaluate . . . . .	2
1.4	ARexx - GetLine . . . . .	2
1.5	ARexx - Help . . . . .	2
1.6	ARexx - Hide . . . . .	2
1.7	ARexx - Info . . . . .	3
1.8	ARexx - OpenEnvironment . . . . .	3
1.9	ARexx - Quit . . . . .	3
1.10	ARexx - Show . . . . .	3
1.11	ARexx - Error codes . . . . .	4
1.12	ARexx - Syntax . . . . .	4

# Chapter 1

## ARexx

### 1.1 ARexx port

ARexx port :

[Click here](#) for the main documentation.

As all MUI applications, Laplace has an ARexx port. The name of the port is LAPLACE}. Some commands are common to all MUI applications, they are marked with an \* :

Argument syntax

Error codes

CloseEnvironment ENV/N/A

Evaluate ENV/N,EXP/K/A

GetLine ENV/N/A,LINE/N/A,E=EXP/S,R=RESULT/S

\* Help FILE/A

\* Hide

\* Info ITEM/A

OpenEnvironment

\* Quit

\* Show

### 1.2 ARexx - CloseEnvironment

CloseEnvironment :

Description :

Close an previously opened environment freeing all resources.

Arguments :

ENV/N/A - the id number of the environment to close

## 1.3 ARexx - Evaluate

Evaluate :

Description :

Evaluate the given expression. The result will be a less or more readable string in the RESULT variable ;-) The return code will be zero for success or an **error code**.

If you give an valid ENV id, than the given expression will be appended at the end of the environment and will be evaluated in this context. You may only give single expressions, if you supply more (seperated by commas), only the first on will be processed.

If you omit the ENV keyword, the expression will be evaluated in the base context, which means that only definitions from InitARexx.lh are available. All definitions done in this expression will be lost and have no effect on later evaluations, you have to open an environment to do this. Several expressions can be seperated by commas, so all needed variables can to be declared in the expression. Only the result of the last expression will be returned.

Arguments :

ENV/N - the id number of the environment

EXP/K/A - expression to evaluate

## 1.4 ARexx - GetLine

GetLine :

Description :

Retrieve an entry from an environment. Lines are numbered 0,1,2,..., every **Evaluate** appends a new line.

If you set the EXP keyword then the expression of the specified line will be returned in the RESULT variable.

If you set the RESULT keyword then the result of the specified line will be returned in the RESULT variable.

Arguments :

ENV/N/A - the id number of the environment

LINE/N/A - ordinal number of line to retrieve

E=EXP/S - see description

R=RESULT/S - see description

## 1.5 ARexx - Help

Help (MUI command) :

Description :

A list of all ARexx commands available for the application is written into the given file.

Arguments :

FILE/A - Filename of the command list

## 1.6 ARexx - Hide

Hide (MUI command) :

Description :

Hides (iconifies) Laplace.

---

## 1.7 ARexx - Info

Info (MUI command) :

Description :

According to the given parameter the result string is filled with the following contents :

title Title of the application (guess what it is...)

author Author of the application (mmmhh... who can this be ??)

copyright Copyright message

description Short description

version Version string

base Name of the ARexx port

screen Name of the public screen

Arguments :

ITEM/A - see description

## 1.8 ARexx - OpenEnvironment

OpenEnvironment :

Description :

This will open an environment to work with. You can add lines using **Evaluate** and retrieve them using **GetLine**. RESULT will be an ID number or zero in case of insufficient memory. Don't forget to call **CloseEnvironment** after usage.

Laplace will try to load and process the include file InitARexx.lh, so all needed definition can be placed there. This works similar to the Init.lh file for windows.

## 1.9 ARexx - Quit

Quit (MUI command) :

Description :

Quit Laplace ;-(

## 1.10 ARexx - Show

Show (MUI command) :

Description :

Shows (pops up) Laplace, if it is iconified.

## 1.11 ARexx - Error codes

Error codes :

In case of an error the return code (RC) will be set to one of the following values :

- 1 : Wrong command definition in host program. Should never happen.
- 2 : Out of memory.
- 3 : Unknown ARexx command.
- 4 : Syntax error.
- 5 : Evaluation error. Error description in RESULT
- 6 : Invalid line number given in **GetLine**.
- 7 : Invalid ENV id.

## 1.12 ARexx - Syntax

Syntax :

ARexx commands of MUI applications usually use the same syntax as DOS commands, so does Laplace. Arguments are specified by keywords, this way you don't have to remember a special order of arguments. Furthermore the keywords are easier to remember than strange shortcuts. The format of the arguments is specified using a special template, e.g.

GetLine ENV/N/A,LINE/N/A,E=EXP/S,R=RESULT/S

which means that the command GetLine accepts four arguments : ENV, LINE, EXP and RESULT. You can use E and R as shortcuts for EXP and RESULT. The argument type is specified by some tags of the form /A, /N...

/A - Required

This means that you must supply this argument, or an error is returned.

/S - Switch

There is no argument for this keyword, setting this keyword 'switches the argument on'.

/T - Toggle

Similar to /S, but 'toggles' the default value of the argument.

/K - Keyword

This introduces a string argument. With a template like Command ITEM/K you should use Command ITEM=Hallo, Command ITEM="Hallo world!" or similar. You may omit the =. If the argument contains spaces you have to use quotes, otherwise quotes are optional.

/N - Number

Similar to /K, but the argument is an integer number like Command LINE=12.

/F - Rest of line

If this is specified, the entire rest of the line is taken as the argument for the option, even if other keywords appear in it.

/M - Multiple strings

Similar to /K, but takes several strings as argument. Any arguments not considered to be part of another keyword will be added to this argument.

A valid command line for the example above could be

GetLine ENV=123 LINE 12 RESULT

---