

PowerData

COLLABORATORS

	<i>TITLE :</i> PowerData		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 28, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PowerData	1
1.1	PowerData ARexx Documentation (26.7.1993)	1
1.2	Introduction	2
1.3	PowerData and ARexx	3
1.4	ARexx Command: VERSION	4
1.5	ARexx Command: TMPFILES	4
1.6	ARexx Command: ENABLE	5
1.7	ARexx Command: DISABLE	5
1.8	ARexx Command: SHOWWINDOW	5
1.9	ARexx Command: HIDEWINDOW	6
1.10	ARexx Command: QUIT	6

Chapter 1

PowerData

1.1 PowerData ARexx Documentation (26.7.1993)

Documentation for

PowerData
Version 38.200

A transparent, application independent
powerpacker file loader and saver for
Workbench 2.04 and above

Written by Michael Berg
Copyright © 1992, 1993 by Michael Berg
All rights reserved.

Subsection: The ARexx Interface

This document describes the ARexx commands currently incorporated into PowerData. Please select any one of the following items:

Brief Introduction to ARexx
General Information

ARexx and PowerData
Further details..

Available ARexx Commands

Version
Return a version string in RESULT

TmpFiles
Returns the current number of active, temporary files

Enable

```
    Enables PowerData (similar to pressing the hotkey)

Disable
    Disables PowerData (similar to pressing the hotkey)

ShowWindow
    Show PowerData's GUI

HideWindow
    Hide PowerData's GUI

Quit
    Kick PowerData off the system
```

1.2 Introduction

Introduction to ARexx

For those of you who haven't got a clue what ARexx is, let me just try to outline the general idea.

When someone writes a program, the programmer implements a number of features in it. These features are all available when using the program. For example, in a drawing program, you would have functions to plot a point when the user clicks somewhere on the screen, plus functions to draw lines, circles, squares, polygons and filled shapes. All these functions can be activated by the user, in an interactive session.

And interactive is indeed the keyword here. Suppose that the user wanted to draw a shape from a huge list of (x,y) coordinates (geographic data, for example). The only way to do this would be for the user to read each coordinate individually, and then use the mouse to accurately click in the point on the screen.

As you can well imagine, this would take almost forever. And even if our poor user ever finishes, the problem will still be there the next time such a situation occurs.

Luckily, there's ARexx. If the programmer is clever enough to implement an ARexx interface in his drawing program, the plotting could be done automatically, assuming the presence of a PLOT command.

Here is an example of an ARexx script that could pull this off:

```
01 /* PlotTest.rexx - plot lots of coordinates stored in a file */
02 /* Author: John Doe */
03 /* Date: 4 feb 1993 */
04
05 /* Talk to the drawing program */
06 address 'SuperDuperDraw'
07
08 /* Open up the coordinates file */
09 if open('inputfile','dhl:coords.dat','r') then do
10     do while ~eof('inputfile')
11         xcoord = readln('inputfile')
```

```
13     ycoord = readln('inputfile')
14
15     /* Now, tell SuperDuperDraw to plot this pair of coordinates */
16     plot xcoord ycoord
17   end
18 end
19 else say 'Cannot open input file!'
```

The numbers to the left (in the beginning of each line) denote line numbers and are for your viewing pleasure only. They would not be entered into the actual ARexx script.

The command in line 6 tells the ARexx server that we want to talk to the program that has an ARexx port named SuperDuperDraw. In a real ARexx script, you would have to check if SuperDuperDraw was actually running or not, but this part is omitted for the sake of clarity and simplicity. In any event, if the ARexx server cannot find the program, the script will abort with an error message.

Next, in line 9, the coordinate file is opened, and read until the end of file marker is reached. The coordinate file is supposed to have one number per line, in pairs of alternating X and Y coordinates.

The command in line 16 does all the magic. There is no "real" ARexx command called PLOT, so the server forwards this command to whoever we are addressing, asking it to process that command. Command names are case insensitive, so it makes no difference if we say "plot" or "PLOT" or even "pLoT".

As it happens, our SuperDuperProgram does have an ARexx command called PLOT, so it responds to the request by plotting the coordinate pair given as parameters to the PLOT command.

By now you must have realized that using the ARexx interface would save our user a lot of work, and also speed up the process tremendously. This is the real advantage of ARexx: It automates sequences of actions that would otherwise have to be carried out interactively. This is also the reason why ARexx scripts are sometimes called macros.

Enough about drawing programs. Let's get back on track and talk about PowerData's ARexx interface. Please go on and read the section on "PowerData and ARexx".

Take me there!

1.3 PowerData and ARexx

As of v38.115 PowerData now comes with basic ARexx support. There are commands to enable and disable it, have it show or hide its window, or make it quit entirely.

For example, this enables enables you to use the AmigaDOS "Copy" command without having to turn PowerData on and off all the time. This is illustrated in the accompanying ARexx example script called "Copy.rexx".

When addressing PowerData, you must use the port name "POWERDATA". Keep in

mind that port names are case sensitive, so always remember to spell the name exactly as shown here, or you will get a message saying "Host environment not found" when you try running your script.

1.4 ARexx Command: VERSION

NAME

VERSION - Give out version information

SYNOPSIS

VERSION

FUNCTION

This command gives out the same kind of version information that you would get if you used the AmigaDOS "Version" command.

INPUTS

None.

OUTPUTS

Returns version string in the RESULT variable.

NOTES

Remember to put "options results" in your script, or you won't get anything back.

SEE ALSO

1.5 ARexx Command: TMPFILES

NAME

TMPFILES - Number of active, temporary files

SYNOPSIS

TMPFILES

FUNCTION

This function returns the number of currently active, temporary files. Temporary files are created by PowerData whenever a program attempts to open a powerpacked file, or attempts to create a new file and the "Crunch" flag is set. This function provides a means of determining the number of temporary files that PowerData is currently handling.

INPUTS

None.

OUTPUTS

Returns number of files in RC

NOTES

SEE ALSO

1.6 ARexx Command: ENABLE

NAME

ENABLE - Turn PowerData on

SYNOPSIS

ENABLE

FUNCTION

If PowerData is currently off, it is reactivated. Otherwise, this function is a no-op. ENABLE is functionally equivalent to activating PowerData by using the appropriate hotkey.

INPUTS

None.

OUTPUTS

None yet.

NOTES

SEE ALSO

DISABLE

1.7 ARexx Command: DISABLE

NAME

DISABLE - Turn PowerData off

SYNOPSIS

DISABLE

FUNCTION

If PowerData is currently on, it is deactivated. Otherwise, this function is a no-op. DISABLE is functionally equivalent to deactivating PowerData by using the appropriate hotkey.

INPUTS

None.

OUTPUTS

None yet.

NOTES

SEE ALSO

ENABLE

1.8 ARexx Command: SHOWWINDOW

NAME

SHOWWINDOW - Make PowerData show its GUI

SYNOPSIS

SHOWWINDOW

FUNCTION

If PowerData's GUI is currently not shown, it will be. Otherwise, this function is a no-op. SHOWWINDOW is functionally equivalent to activating the GUI by using the appropriate hotkey.

INPUTS

None.

OUTPUTS

None yet.

NOTES

SEE ALSO

HIDEWINDOW

1.9 ARexx Command: HIDEWINDOW

NAME

SHOWWINDOW - Make PowerData hide its GUI

SYNOPSIS

SHOWWINDOW

FUNCTION

If PowerData's GUI is currently shown, it will be hidden. Otherwise, this function is a no-op. HIDEWINDOW is functionally equivalent to clicking on the closewindow gadget in PowerData's GUI window.

INPUTS

None.

OUTPUTS

None yet.

NOTES

SEE ALSO

SHOWWINDOW

1.10 ARexx Command: QUIT

NAME

QUIT - Tell PowerData to close up shop

SYNOPSIS

QUIT

FUNCTION

This function will make PowerData terminate. It does not matter if PowerData's GUI is currently hidden, or if PowerData is disabled.

INPUTS

None.

OUTPUTS

None yet.

NOTES

SEE ALSO