

Arexx

Copyright © 1992-1996 Nova Design, Inc.

COLLABORATORS

	<i>TITLE :</i> Arexx	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		January 13, 2023
		<i>SIGNATURE</i>

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Arexx	1
1.1	ImageFX Arexx Documentation Contents	1
1.2	Document Format	1

Chapter 1

Arexx

1.1 ImageFX Arexx Documentation Contents

1. General Information:

- Document Format

2. ImageFX:

- ImageFX Commands

- Hook Commands

- Scanner Commands

- Render Commands

- Printer Commands

- Loader Arguments

- Saver Arguments

3. CineMorph:

- CineMorph Commands

1.2 Document Format

IMAGEFX AREXX COMMANDS
RELEASE 2.5

This document assumes that you are familiar with programming in the Arexx language, and only describes the commands specific to ImageFX.

All commands, unless otherwise noted, return a non-zero error code if something went wrong. This includes cancelling the operation. If the return code is 0, then a result string may be provided.

Those commands marked with (Arexx Only) may only be used properly from an Arexx macro. All other commands may be used in scripts, key

definitions, toolbox definitions, the command shell, or Arexx macros at will.

When a command is invoked from the shell, any result string returned will be printed to the console. Arexx errors will also be displayed on the console.

Commands and arguments marked with (2.0) are new for ImageFX 2.0.

Commands and arguments marked with (2.1) are new for ImageFX 2.1.

Commands and arguments marked with (2.1a) are new for ImageFX 2.1a.

Commands and arguments marked with (2.5) are new for ImageFX 2.5.

HOW TO READ THIS DOCUMENT

FORMAT

This gives the command name followed by its argument template. The argument template follows the standard AmigaDOS CLI command format (see your AmigaDOS manual for more details about command templates). Each keyword (separated by commas) represents a possible argument. The keyword is followed by a / and a letter to indicate what the keyword's parameter type (numeric, switch, etc.) is. Most of the time, you do not need to give the keyword itself.

Here are the types of arguments:

```
/A Always required (cumulative with other modifiers).
/F Final argument (rest of line).
/K Keyword required.
/N Number argument.
/S Switch keyword.
```

Example #1:

FORMAT

```
RequestFile Title/A,Path,File,Pattern
```

There are four possible arguments to this command. The first one MUST be specified. The remaining three are optional. All are text arguments. This template can be filled in any of the following ways:

```
/* the complete longhand format: */
```

```
RequestFile TITLE "Load Image" PATH "DH0:" FILE "Image" PATTERN
"#?.pic"
```

```
/* the abbreviated format (if the keywords are not specified, the
arguments are filled in in order.) */
```

```
RequestFile "Load Image" "DH0:" "Image" "#?.pic"
```

```
/* only the first argument is required: */
```

```
RequestFile "Load Image"
```

```
/* to skip arguments... you MUST use the "pattern" keyword in this
case otherwise the "#?.pic" will go into the "path" argument: */
```

```
RequestFile "Load Image" PATTERN "#?.pic"
```

```
/* quotation marks around arguments are only necessary if there are
spaces in the argument. */
```

FUNCTION

Describes what this command actually does.

INPUTS

Describes in detail all of the possible arguments to the command.

Note that any arguments containing spaces must be surrounded by quotes. Also note that you must surround negative numbers with quotes, too, or Arexx will think you are trying to do an arithmetic operation.

RESULT

Describes the result string returned from the command, if one is in fact returned. If the command is used from Arexx, this result string will be found in the RESULT variable. If the command is typed in the command shell, then the result string is just printed out to the console. The result string is lost when invoked from a key, script, or the toolbox.

SEE ALSO

Possible related commands.
