

English/FileXARexx

COLLABORATORS

	TITLE : English/FileXARexx		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		December 8, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	English/FileXARexx	1
1.1	English/FileXARexx.guide	1
1.2	FileXARexx.guide/Introduction	1
1.3	FileXARexx.guide/Command overview	2
1.4	FileXARexx.guide/ACTIVATE	4
1.5	FileXARexx.guide/ACTIVATEVIEW	4
1.6	FileXARexx.guide/ACTIVATEWINDOW	4
1.7	FileXARexx.guide/ADJUSTWINDOWSIZE	5
1.8	FileXARexx.guide/APPEND	5
1.9	FileXARexx.guide/ASSIGNCOMMAND	6
1.10	FileXARexx.guide/CHANGEWINDOW	6
1.11	FileXARexx.guide/CLEAR	7
1.12	FileXARexx.guide/CLOSE	7
1.13	FileXARexx.guide/COLUMN	8
1.14	FileXARexx.guide/COPY	8
1.15	FileXARexx.guide/CURSOR	9
1.16	FileXARexx.guide/CUT	9
1.17	FileXARexx.guide/DEACTIVATE	10
1.18	FileXARexx.guide/EXECUTECOMMAND	10
1.19	FileXARexx.guide/EXPANDVIEW	10
1.20	FileXARexx.guide/FAULT	11
1.21	FileXARexx.guide/FILL	11
1.22	FileXARexx.guide/FIND	12
1.23	FileXARexx.guide/FINDCHANGE	13
1.24	FileXARexx.guide/FINDNEXT	14
1.25	FileXARexx.guide/FONT	15
1.26	FileXARexx.guide/GETATTR	15
1.27	FileXARexx.guide/GETBLOCK	16
1.28	FileXARexx.guide/GETBYTES	16
1.29	FileXARexx.guide/GOTOBOOKMARK	17

1.30 FileXARexx.guide/GOTOBYTE	17
1.31 FileXARexx.guide/GOTOCOLUMN	18
1.32 FileXARexx.guide/GOTOLINE	18
1.33 FileXARexx.guide/GRABMEMORY	18
1.34 FileXARexx.guide/GROWVIEW	19
1.35 FileXARexx.guide/HELP	19
1.36 FileXARexx.guide/INSERT	20
1.37 FileXARexx.guide/LINE	20
1.38 FileXARexx.guide/LOADBLOCK	21
1.39 FileXARexx.guide/LOADCOMMANDS	21
1.40 FileXARexx.guide/LOADDISPLAYTYP	22
1.41 FileXARexx.guide/LOADSETTINGS	22
1.42 FileXARexx.guide/LOCKGUI	22
1.43 FileXARexx.guide/MOVEWINDOW	23
1.44 FileXARexx.guide/NEW	23
1.45 FileXARexx.guide/NEXT	24
1.46 FileXARexx.guide/NOP	24
1.47 FileXARexx.guide/OPEN	24
1.48 FileXARexx.guide/PASTE	25
1.49 FileXARexx.guide/POSITION	25
1.50 FileXARexx.guide/PREVIOUS	26
1.51 FileXARexx.guide/PRINT	26
1.52 FileXARexx.guide/PRINTBLOCK	27
1.53 FileXARexx.guide/QUIT	27
1.54 FileXARexx.guide/REDO	27
1.55 FileXARexx.guide/REQUESTFILE	28
1.56 FileXARexx.guide/REQUESTNOTIFY	28
1.57 FileXARexx.guide/REQUESTRESPONSE	29
1.58 FileXARexx.guide/REQUESTSTRING	29
1.59 FileXARexx.guide/RX	30
1.60 FileXARexx.guide/SAVE	30
1.61 FileXARexx.guide/SAVEAS	30
1.62 FileXARexx.guide/SAVEBLOCK	31
1.63 FileXARexx.guide/SAVECOMMANDS	31
1.64 FileXARexx.guide/SAVECOMMANDSAS	32
1.65 FileXARexx.guide/SAVESETTINGS	32
1.66 FileXARexx.guide/SAVESETTINGSAS	32
1.67 FileXARexx.guide/SETATTR	33
1.68 FileXARexx.guide/SETBLOCK	33

1.69	FileXARexx.guide/SETBLOCKMARK	34
1.70	FileXARexx.guide/SETBOOKMARK	34
1.71	FileXARexx.guide/SETBYTES	35
1.72	FileXARexx.guide/SETSCREEN	35
1.73	FileXARexx.guide/SHRINKVIEW	36
1.74	FileXARexx.guide/SIZEWINDOW	36
1.75	FileXARexx.guide/SPLIT	36
1.76	FileXARexx.guide/UNDO	37
1.77	FileXARexx.guide/UNLOCKGUI	37
1.78	FileXARexx.guide/WINDOWTOBACK	38
1.79	FileXARexx.guide/WINDOWTOFRONT	38
1.80	FileXARexx.guide/ZIPWINDOW	38
1.81	FileXARexx.guide/Attributes	38

Chapter 1

English/FileXARexx

1.1 English/FileXARexx.guide

```
$VER: FileXARexxDocumentation 2.0 (29.5.94)
```

```
FileX - A binary editor
```

```
© Copyright 1993-1994 by Klaas Hermanns
```

```
Shareware, all rights reserved.
```

```
Introduction
```

```
Command overview
```

```
Attributes
```

1.2 FileXARexx.guide/Introduction

```
Introduction
```

```
*****
```

FileX has an arexx port which is based on the Commodore Amiga User Interface Style Guide.

If there is nothing different specified for the PORTNAME at the beginning, the port will be named 'FILEX'. In case that this port name does already exist, a number is being added to the name and increased until a not already used name is found. ('NAME' -> 'NAME.1' -> 'NAME.2')

The arguments will be evaluated in the way AmigaDOS does. If the different argument types are unknown, the following list can be used as a reference.

<arg>

A simple string. It's not necessary to specify the argument's name.

<arg>/S
Switch. Will be set through declaring an argument.

<arg>/K
Key word. This argument must be given with an argument's name.
(`<arg> = value' oder '<arg> value'.

<arg>/N
Number. This argument must be a decimal number.

<arg>/T
Changing switch. Through specification of this argument a value
will change its condition.

<arg>/A
This argument must be given.

<arg>/F
The rest of the line will be understood as an argument, although if
there are some more keywords

<arg>/M
Serveral values are possible.

For all commands that return something it's possible to set a
variable to the return value by specifying the name of it with the
argument VAR. The same applies to the argument STEM and its
stemvariables.

All commands with an ID argument return rc = 5, if the ID is
invalid. If there is no ID given the current file/view/window is used
for the command.

1.3 FileXARexx.guide/Command overview

Command overview

Name:	Argumente => Rückgabewerte:
ACTIVATE	
ACTIVATEVIEW	VIEWID=VID/K/N
ACTIVATEWINDOW	WINDOWID=WID/K/N
ADJUSTWINDOWSIZE	WINDOWID=WID/K/N
APPEND	VIEWID=VID/K/N
ASSIGNCOMMAND	NUMBER/K/N, NAME/K
CHANGEWINDOW	LEFTEDGE/K/N, TOPEDGE/K/N, HEIGHT/K/N, WIDTH/K/N, WINDOWID=WID/K ↵
/N	
CLEAR	FORCE/S, VIEWID=VID/K/N
CLOSE	VIEW/S, WINDOW/S, FORCE/S, VIEWID=VID=WINDOWID=WID/K/N
COLUMN	DELTA/N/A, VIEWID=VID/K/N
COPY	VIEWID=VID/K/N
CURSOR	UP/S, DOWN/S, LEFT/S, RIGHT/S, VIEWID=VID/K/N
CUT	VIEWID=VID/K/N

DEACTIVATE	
EXECUTECOMMAND	NUMBER/K/N, NAME/K
EXPANDVIEW	VIEWID=VID/K/N
FAULT	VAR/K, STEM/K, NUMBER/N/A => DESCRIPTION
FILL	VIEWID=VID/K/N
FIND	PROMPT/S, HEX/S, BACKWARDS/S, QUIET/S, TEXT/F, VIEWID=VID/K/N
FINDCHANGE	PROMPT/S, HEX/S, ALL/S, BACKWARDS/S, QUIET/S, FIND, CHANGE, VIEWID= ↔
VID/K/N	
FINDNEXT	BACKWARDS/S, QUIET/S, VIEWID=VID/K/N
FONT	NAME/A, SIZE/N/A, WINDOWID=WID/K/N
GETATTR	OBJECT/A, FIELD, ID/K/N, STEM/K, VAR/K
GETBLOCK	VAR/K, STEM/K => BLOCK
GETBYTES	VAR/K, STEM/K, POS/N/A, NUMBER/N/A, FILEID=FID/K/N => HEXSTRING
GOTOBOOKMARK	NUMBER/N, VIEWID=VID/K/N
GOTOBYTE	POS/N/A, VIEWID=VID/K/N
GOTOCOLUMN	COLUMN/N/A, VIEWID=VID/K/N
GOTOLINE	LINE/N/A, VIEWID=VID/K/N
GRABMEMORY	START/N/A, END/N/A, FORCE/S, VIEWID=VID/K/N
GROWVIEW	VIEWID=VID/K/N
HELP	VAR/K, STEM/K, COMMAND => COMMANDDESC, COMMANDLIST/M
INSERT	VIEWID=VID/K/N
LINE	DELTA/N/A, VIEWID=VID/K/N
LOADBLOCK	NAME
LOADCOMMANDS	NAME
LOADDISPLAYTYP	NAME
LOADSETTINGS	NAME
LOCKGUI	
MOVEWINDOW	LEFTEDGE/N, TOPEDGE/N, WINDOWID=WID/K/N
NEW	VIEW/S, WINDOW/S
NEXT	VIEW/S, WINDOW/S
NOP	
OPEN	FILENAME, FORCE/S, VIEWID=VID/K/N
PASTE	VIEWID=VID/K/N
POSITION	SOF/S, EOF/S, SOL/S, EOL/S, VIEWID=VID/K/N
PREVIOUS	VIEW/S, WINDOW/S
PRINT	FILEID=FID/K/N
PRINTBLOCK	
QUIT	FORCE/S
REDO	NUMBER/N, FILEID=FID/K/N
REQUESTFILE	VAR/K, STEM/K, TITLE/K, PATH/K, FILE/K, PATTERN/K => FILENAME
REQUESTNOTIFY	PROMPT
REQUESTRESPONSE	BUTTONS/K, PROMPT/K
REQUESTSTRING	VAR/K, STEM/K, PROMPT/K, DEFAULT/K => STRING
RX	VAR/K, STEM/K, CONSOLE/S, ASYNC/S, COMMAND/F => RC/N, RESULT
SAVE	FILEID=FID/K/N
SAVEAS	NAME, FILEID=FID/K/N
SAVEBLOCK	NAME
SAVECOMMANDS	
SAVECOMMANDSAS	NAME
SAVESETTINGS	
SAVESETTINGSAS	NAME
SETATTR	OBJECT/A, FIELD, ID/K/N, FROMSTEM/K, FROMVAR/K
SETBLOCK	BLOCK/A
SETBLOCKMARK	VIEWID=VID/K/N
SETBOOKMARK	NUMBER/N, VIEWID=VID/K/N
SETBYTES	POS/N/A, HEXSTRING/A, FILEID=FID/K/N
SETSCREEN	NAME/K, OWN/S

SHRINKVIEW	VIEWID=VID/K/N
SIZEWINDOW	HEIGHT/N,WIDTH/N,WINDOWID=WID/K/N
SPLIT	VIEW/S,WINDOW/S
UNDO	NUMBER/N,FILEID=FID/K/N
UNLOCKGUI	
WINDOWTOBACK	WINDOWID=WID/K/N
WINDOWTOFRONT	WINDOWID=WID/K/N
ZIPWINDOW	WINDOWID=WID/K/N

1.4 FileXARexx.guide/ACTIVATE

ACTIVATE
=====

Synopsis:
 ACTIVATE

Function:
 Reactivates FileX from iconify.

Example:

```
DEACTIVATE      /* Equivalent to menu: 'Project/Iconify' */

/* Now an icon will appear on the workbench. FileX 'sleeps'. */

ACTIVATE        /* Open FileX with all windows again */
```

See also:
 DEACTIVATE

1.5 FileXARexx.guide/ACTIVATEVIEW

ACTIVATEVIEW
=====

Synopsis:
 ACTIVATEVIEW VIEWID=VID/K/N

Function:
 Activates an view.

Arguments:
 VIEWID=VID/K/N:
 ID of a view that should be activated.

1.6 FileXARexx.guide/ACTIVATEWINDOW

ACTIVATEWINDOW
=====

Synopsis:

ACTIVATEWINDOW WINDOWID=WID/K/N

Function:

Activates an edit window.

Arguments:

WINDOWID=WID/K/N:

ID of a edit window that should be activated.

1.7 FileXARexx.guide/ADJUSTWINDOWSIZE

ADJUSTWINDOWSIZE
=====

Synopsis:

ADJUSTWINDOWSIZE WINDOWID=WID/K/N

Function:

Adjusts the window to the size of edit views.

Arguments:

WINDOWID=WID/K/N:

ID of a window.

See also:

Menu: 'Settings/Adjust window size'

1.8 FileXARexx.guide/APPEND

APPEND
=====

Synopsis:

APPEND VIEWID=VID/K/N

Function:

Appends the current block to the end of a file.

Arguments:

VIEWID=VID/K/N:

ID of a view that belongs to the file.

See also:

Menu: 'Edit/Append'

1.9 FileXARexx.guide/ASSIGNCOMMAND

ASSIGNCOMMAND

=====

Synopsis:

ASSIGNCOMMAND NUMBER/K/N,NAME/K

Function:

Set one of the commands in the arexx menu.

Arguments:

NUMBER/K/N:

Number between 0 and 10 which defines the command's position.
A requester will appear if no number has been specified.

NAME/K:

Command's file name. If no name is given a file requester
will appear.

Example:

ASSIGNCOMMAND 0 'dh5:myarexxcommands/filexcommand'

See also:

Menu: 'ARexx/Assign command', LOADCOMMANDS, SAVECOMMANDS,
SAVECOMMANDSAS

1.10 FileXARexx.guide/CHANGEWINDOW

CHANGEWINDOW

=====

Synopsis:

CHANGEWINDOW

LEFTEDGE/K/N, TOPEDGE/K/N, HEIGHT/K/N, WIDTH/K/N, WINDOWID=WID/K/N

Function:

Changes the position and/or size of a edit window.

Arguments:

LEFTEDGE/K/N:

New left offset.

TOPEDGE/K/N:

New top offset.

HEIGHT/K/N:

New height.

WIDTH/K/N:

Neq width.

WINDOWID=WID/K/N

ID of a edit window that should be changed.

If one of these arguments is not given, it is not changed.

Example:

```
/* Window to maximum size */  
CHANGEWINDOW LEFTEDGE 0 TOPEDGE 0 HEIGHT 4000 WIDTH 4000
```

See also:

ZIPWINDOW, MOVEWINDOW, SIZEWINDOW, Attributes: 'WINDOW/#?'

1.11 FileXARexx.guide/CLEAR

CLEAR

=====

Synopsis:

CLEAR FORCE/S,VIEWID=VID/K/N

Function:

Clears a view.

Arguments:

FORCE/S:

The changes requester will be suppressed.

VIEWID=VID/K/N

ID of a view that should be cleared.

Return codes:

rc = 5, if the requester is cancelled.

See also:

Menu: 'Project/New'

1.12 FileXARexx.guide/CLOSE

CLOSE

=====

Synopsis:

VIEW/S,WINDOW/S,FORCE/S,VIEWID=VID=WINDOWID=WID/K/N

Function:

Closes a view or a window.

Arguments:

VIEW/S:

The ID is an view ID.

WINDOW/S:

The ID is an window ID.

FORCE/S:

The changes requester will be suppressed.

VIEWID=VID=WINDOWID=WID/K/N:

ID of a view resp. window that should be closed.

See also:

QUIT

1.13 FileXARexx.guide/COLUMN

COLUMN

=====

Synopsis:

COLUMN DELTA/N/A,VIEWID=VID/K/N

Function:

Moves the cursor to the left or to the right.

Arguments:

DELTA/N/A:

Number of steps. Positive numbers move the cursor to the right.

VIEWID=VID/K/N

ID of a view.

See also:

GOTOBYTE, GOTOCOLUMN, GOTOLINE, LINE

1.14 FileXARexx.guide/COPY

COPY

=====

Synopsis:

COPY VIEWID=VID/K/N

Function:

Copies the current marked block to the block buffer.

Arguments:

VIEWID=VID/K/N

ID of a view.

See also:

Menu: `Edit/Copy`, SETBLOCKMARK

1.15 FileXARexx.guide/CURSOR

CURSOR

=====

Synopsis:

CURSOR UP/S,DOWN/S,LEFT/S,RIGHT/S,VIEWID=VID/K/N

Function:

Move the cursor.

Arguments:

UP/S:

Moves the cursor one step up.

DOWN/S:

Moves the cursor one step down.

LEFT/S:

Moves the cursor one step to the left.

RIGHT/S:

Moves the cursor one step to the right.

VIEWID=VID/K/N:

ID of a view.

UP and DOWN as well as LEFT and RIGHT exclude each other.

See also:

GOTOBYTE, LINE, COLUMN

1.16 FileXARexx.guide/CUT

CUT

===

Synopsis:

CUT VIEWID=VID/K/N

Function:

Cuts the current marked block out.

Arguments:

VIEWID=VID/K/N:

ID of a view.

See also:

Menu: 'Edit/Cut', SETBLOCKMARK

1.17 FileXARexx.guide/DEACTIVATE

DEACTIVATE
=====

Synopsis:
DEACTIVATE

Function:
Iconifies FileX. The window will be closed and an icon will appear on the workbench.

ATTENTION!: All arexx comannnds which output something in an edit window or change an edit window in any way, can lead to a crash during the iconify.

See also:
ACTIVATE

1.18 FileXARexx.guide/EXECUTECOMMAND

EXECUTECOMMAND
=====

Synopsis:
EXECUTECOMMAND NUMBER/K/N, NAME/K

Function:
Executes an arexx command.

Arguments:
NUMBER/K/N:
Number between 0 and 9 of an with ASSIGNCOMMAND installed command.

NAME/K:
Name of an arexx script. Both arguments exclude each other.

See also:
ASSIGNCOMMAND

1.19 FileXARexx.guide/EXPANDVIEW

EXPANDVIEW
=====

Synopsis:

EXPANDVIEW VIEWID=VID/K/N

Function:

Expands the current view to its maximum size. All other views in the window will become one line height.

Arguments:

VIEWID=VID/K/N:
ID of a view.

See also:

SHRINKVIEW, GROWVIEW

1.20 FileXARexx.guide/FAULT

FAULT
=====

Synopsis:

FAULT VAR/K,STEM/K,NUMBER/N/A => DESCRIPTION

Function:

Determines the text belonging to an dos error code.

Arguments:

NUMBER/N/A:
Dos error code.

Return codes:

DESCRIPTION:
The error text.

Example:

FAULT 114

```
/* Result value using localized workbench: */  
/* ': Falsches Namensmuster' */
```

1.21 FileXARexx.guide/FILL

FILL
=====

Synopsis:

FILL VIEWID=VID/K/N

Function:

Fills the current marked block with the curent fill string.

Arguments:

VIEWID=VID/K/N:

ID of a view.

Example:

```
/* If there is a block already marked, */
/* then neutralize it. */

GETATTR VIEW MARK
if result=TRUE then SETBLOCKMARK

/* Set fillstring to 'Fill fillaround!' */

SETATTR APPLICATION FILLSTRING FROMVAR "Fill fillaround!"

/* Set cursor to the fill-area's beginning */
GOTOBYTE 10

/* Set block mark */
SETBLOCKMARK

/* Move cursor */
GOTOBYTE 100

/* Fill the area */
FILL

/* Remove block marking */
SETBLOCKMARK
```

See also:

Menu: 'Edit/Fill', SETBLOCKMARK, SETATTR

1.22 FileXARexx.guide/FIND

FIND

====

Synopsis:

FIND PROMPT/S,HEX/S,BACKWARDS/S,QUIET/S,TEXT/F,VIEWID=VID/K/N

Function:

Searches a string or a hex string within the current file. By success, the cursor is being positioned at the string's beginning. The search begins at the current cursor position.

Arguments:

PROMPT/S:

Opens the searchrequester. BACKWARDS and QUIET will be ignored.

HEX/S:
Hex search.

BACKWARDS/S:
Search backwards.

QUIET/S:
No requester if the search fails.

TEXT/F:
The search string or the hex search string.

VIEWID=VID/K/N:
ID of a view.

Return codes:
rc = 5, if the search fails.

Example:

```
/* Search the string '$VER:' without opening */
/* a requester by failure */
FIND QUIET TEXT '$VER:'
```

See also:
Attributes: 'APPLICATION/#?SEARCH', FINDNEXT, FINDCHANGE, Menu:
'Search&Replace/#?'

1.23 FileXARexx.guide/FINDCHANGE

FINDCHANGE
=====

Synopsis:
FINDCHANGE
PROMPT/S, HEX/S, ALL/S, BACKWARDS/S, QUIET/S, FIND, CHANGE, VIEWID=VID/K/N

Function:
Searches a string or a hexstring within a file and replaces it through a new one. Maybe a requester appears to verify the replacement. The search will start from the current cursorposition.

Arguments:
PROMPT/S:
Opens the search requester. ALL, BACKWARDS and QUIET will be ignored. ignoriert.

HEX/S:
Hex search.

ALL/S:
All found positions will be replaced without requester each time.

BACKWARDS/S:
 Search backwards.

QUIET/S:
 No requester if the search fails.

FIND:
 Search text or hex string.

CHANGE:
 Replace text or hex string.

VIEWID=VID/K/N:
 ID of a view.

Return codes:
 rc = 5, if the search fails.

Example:

```
/* Replaces 'Claas' by 'Klaas' */  
FINDCHANGE QUIET ALL 'Claas' 'Klaas'  
FINDCHANGE BACKWARDS QUIET ALL 'Claas' 'Klaas'
```

See also:
 FINDNEXT, FIND, Menu: 'Search&Replace/#?'

1.24 FileXARexx.guide/FINDNEXT

FINDNEXT
=====

Synopsis:
 FINDNEXT BACKWARDS/S, QUIET/S, VIEWID=VID/K/N

Function:
 Continues the last search operation, i.e. either search or search&replace.

Arguments:
 BACKWARDS/S:
 Continue search backwards.

 QUIET/S:
 No requester if the search fails.

 VIEWID=VID/K/N:
 ID of a view.

Return codes:
 rc = 5, if search fails.

See also:
 FIND, FINDCHANGE

1.25 FileXARexx.guide/FONT

FONT
=====

Synopsis:

FONT NAME/A,SIZE/N/A,WINDOWID=WID/K/N

Function:

Sets the font of an edit window.

Arguments:

NAME/A:
Font name.

SIZE/N/A:
Font size.

WINDOWID=WID/K/N:
ID of a window.

Example:

FONT 'horror.font' 8

1.26 FileXARexx.guide/GETATTR

GETATTR
=====

Synopsis:

GETATTR OBJECT/A,FIELD,ID/K/N,STEM/K,VAR/K

Function:

Gets the value of one or more variables used in the program.

Arguments:

OBJECT/A:
Name of the object.

FIELD:
Name of the field.

ID/K/N:
Possible with the objects WINDOW, VIEW and FILE. If not given,
the current view/window/file is used to determine the value.

If FIELD is not given, a STEM variable must be specified which
will be filled with all values of the object.

Example:

```
GETATTR WINDOW TITLE

/* Result contains now the titel of the window */
/* Not very interesting, but who cares :-) */
```

See also:

Attributes, SETATTR

1.27 FileXARexx.guide/GETBLOCK

GETBLOCK
=====

Synopsis:

GETBLOCK VAR/K,STEM/K => BLOCK

Function:

Determines the contents of the block buffer, i.e. the contents
clipboard.

Return codes:

BLOCK:
The block buffer as a hex string.

See also:

SETBLOCK

1.28 FileXARexx.guide/GETBYTES

GETBYTES
=====

Synopsis:

GETBYTES VAR/K,STEM/K,POS/N/A,NUMBER/N/A,FILEID=FID/K/N =>
HEXSTRING

Function:

Determines the value of a number of bytes at a position in the
file.

Arguments:

POS/N/A:
Start of the block.

NUMBER/N/A:
Number of bytes to get.

Return codes:

HEXSTRING:

Block as a hex string.

FILEID=FID/K/N:
ID of a file.

See also:
SETBYTES

1.29 FileXARexx.guide/GOTOBOOKMARK

GOTOBOOKMARK
=====

Synopsis:
GOTOBOOKMARK NUMBER/N, VIEWID=VID/K/N

Function:
Sets the cursor to a position previously marked with SETBOOKMARK.

Arguments:
NUMBER/N/A:
Number between 1 and 10.

VIEWID=VID/K/N:
ID of a view.

See also:
SETBOOKMARK

1.30 FileXARexx.guide/GOTOBYTE

GOTOBYTE
=====

Synopsis:
GOTOBYTE POS/N/A, VIEWID=VID/K/N

Function:
Set the cursor to a specified position.

Arguments:
POS/N/A:
Offset to the start of the file.

VIEWID=VID/K/N:
ID of a view.

See also:
COLUMN, GOTOCOLUMN, GOTOLINE, LINE

1.31 FileXARexx.guide/GOTOCOLUMN

GOTOCOLUMN
=====

Synopsis:
GOTOCOLUMN COLUMN/N/A,VIEWID=VID/K/N

Function:
Sets the cursor to a specified column.

Arguments:
COLUMN/N/A:
Column number.

VIEWID=VID/K/N:
ID of a view.

See also:
COLUMN, GOTOBYTE, GOTOLINE, LINE

1.32 FileXARexx.guide/GOTOLINE

GOTOLINE
=====

Synopsis:
GOTOLINE LINE/N/A,VIEWID=VID/K/N

Function:
Sets the cursor to a specified line.

Arguments:
LINE/N/A:
Line number.

VIEWID=VID/K/N:
ID of a view.

See also:
COLUMN, GOTOBYTE, GOTOCOLUMN, LINE

1.33 FileXARexx.guide/GRABMEMORY

GRABMEMORY
=====

Synopsis:
GRABMEMORY START/N/A,END/N/A,FORCE/S,VIEWID=VID/K/N

Function:

Displays a memory area instead of a file.

Arguments:**START/N/A:**

Start of the memory area.

END/N/A:

End of the memory area.

FORCE/S:

If the current file is changed, the requester will be suppressed.

Return codes:

rc = 5, if the requester is cancelled.

Example:

```
GRABMEMORY 200000 280000 FORCE
```

See also:

Menu: 'Project/Grab memory'

1.34 FileXARexx.guide/GROWVIEW

GROWVIEW

=====

Synopsis:

```
GROWVIEW VIEWID=VID/K/N
```

Function:

If possible the the view grows one line.

Arguments:**VIEWID=VID/K/N:**

ID of a view.

See also:

SHRINKVIEW, EXPANDVIEW

1.35 FileXARexx.guide/HELP

HELP

=====

Synopsis:

```
HELP VAR/K, STEM/K, COMMAND => COMMANDDESC, COMMANDLIST/M
```

Function:

Finds out either an overview of all specified arexx commands or the synopsis of a command.

Arguments:

COMMAND:

Commands name, whose synopsis shall be determined.

If COMMAND is not specified, a command overview will be returned.

Return codes:

If COMMAND is given:

COMMANDDESC:

Synopsis of the command. Else:

COMMANDLIST/M:

Number of commands and overview of all commands.

1.36 FileXARexx.guide/INSERT

INSERT

=====

Synopsis:

INSERT VIEWID=VID/K/N

Function:

Insert the current block of the block buffer at the current cursor position.

Arguments:

VIEWID=VID/K/N:

ID of a view.

See also:

Menu: 'Edit/Insert', SETBLOCKMARK

1.37 FileXARexx.guide/LINE

LINE

=====

Synopsis:

LINE DELTA/N/A,VIEWID=VID/K/N

Function:

Moves the cursor a specified number of lines up or down.

Arguments:

LINE/N/A:

Number of lines. Positive numbers move the cursor down.

```
VIEWID=VID/K/N:  
    ID of a view.
```

See also:

```
COLUMN, GOTOBYTE, GOTOCOLUMN, GOTOLINE
```

1.38 FileXARexx.guide/LOADBLOCK

```
LOADBLOCK  
=====
```

Synopsis:

```
LOADBLOCK NAME
```

Function:

Loads a block into the block buffer(clipboard).

Arguments:

NAME:

Block name. If not given, a file requester will pop up.

Return codes:

rc = 5, if the block couldn't be loaded.

See also:

```
SAVEBLOCK
```

1.39 FileXARexx.guide/LOADCOMMANDS

```
LOADCOMMANDS  
=====
```

Synopsis:

```
LOADCOMMANDS NAME
```

Function:

Loads a set of arexx comannds.

Arguments:

NAME:

If not given, a file requester pops up.

Return codes:

rc = 5, if the file couldn't be loaded.

See also:

Menu: 'ARexx/Load commands', SAVECOMMANDS, SAVECOMMANDSAS

1.40 FileXARexx.guide/LOADDISPLAYTYP

LOADDISPLAYTYP
=====

Synopsis:

LOADDISPLAYTYP NAME

Function:

Loads the user display typ.

Arguments:

NAME:

If not given, a file requester will pop up.

Return codes:

rc = 5, if the file couldn't be loaded.

See also:

Menu: 'Settings/Display/#?'

1.41 FileXARexx.guide/LOADSETTINGS

LOADSETTINGS
=====

Synopsis:

LOADSETTINGS NAME

Function:

Loads a settings file.

Arguments:

NAME:

If not given, a file requester will pop up.

Return codes:

rc = 5, if the file couldn't be loaded.

See also:

Menu: 'Settings/Load Settings...'

1.42 FileXARexx.guide/LOCKGUI

LOCKGUI
=====

Synopsis:

LOCKGUI

Function:

Locks all edit windows. The user can't make any input. Just size and position of the windows can be modified.

Return codes:

rc = 5, if the windows cannot be locked. This is possible when there's still a requester open.

See also:

LOCKGUI

1.43 FileXARexx.guide/MOVEWINDOW

MOVEWINDOW

=====

Synopsis:

MOVEWINDOW LEFTEDGE/N, TOPEDGE/N, WINDOWID=WID/K/N

Function:

Moves the edit window.

Arguments:

LEFTEDGE/N:

New left offset.

TOPEDGE/N:

New top offset.

WINDOWID=WID/K/N:

ID of a window.

1.44 FileXARexx.guide/NEW

NEW

===

Synopsis:

NEW VIEW/S, WINDOW/S

Function:

Opens a new view(and window).

Arguments:

VIEW/S:

The current view is divided into two new views.

WINDOW/S:

Opens a new edit window.

See also:

Menu: `Project/(View|Window)/New`, SPLIT

1.45 FileXARexx.guide/NEXT

NEXT

====

Synopsis:

NEXT VIEW/S,WINDOW/S

Function:

Activates the next view/window.

Arguments:

VIEW/S:

The next view in the current window is activated.

WINDOW/S:

The next window is activated.

See also:

PREVIOUS

1.46 FileXARexx.guide/NOP

NOP

===

Synopsis:

NOP

Function:

This command does nothing.

1.47 FileXARexx.guide/OPEN

OPEN

====

Synopsis:

OPEN FILENAME, FORCE/S, VIEWID=VID/K/N

Function:

Opens a file.

Arguments:

FILENAME:

If not given, a file requester will pop up. Auswahl.

FORCE/S:

The changes requester will be suppressed.

VIEWID=VID/K/N:

ID of a view.

Return codes:

rc = 5, if the file can't be loaded.

See also:

CLEAR, SAVE, SAVEAS

1.48 FileXARexx.guide/PASTE

PASTE

=====

Synopsis:

PASTE VIEWID=VID/K/N

Function:

Pastes the current block into the file at the current cursor position.

Arguments:

VIEWID=VID/K/N:

ID of a view.

See also:

Menu: 'Edit/Paste', SETBLOCKMARK

1.49 FileXARexx.guide/POSITION

POSITION

=====

Synopsis:

POSITION SOF/S, EOF/S, SOL/S, EOL/S, VIEWID=VID/K/N

Function:

Sets the cursor.

Arguments:

SOF/S:

Start of file.

EOF/S:

End of file.

SOL/S:
Start of line.

EOL/S:
End of line.

VIEWID=VID/K/N:
ID of a view.

As well SOF and EOF as SOL and EOL exclude each other.

1.50 FileXARexx.guide/PREVIOUS

PREVIOUS
=====

Synopsis:
PREVIOUS VIEW/S,WINDOW/S

Function:
Activate the previous view resp. edit window.

Arguments:
VIEW/S:
The previous view should be activated.

WINDOW/S:
The previous window should be activated.

See also:
NEXT

1.51 FileXARexx.guide/PRINT

PRINT
=====

Synopsis:
PRINT FILEID=FID/K/N

Function:
Prints the whole file.

Arguments:
ID of a file.

1.52 FileXARexx.guide/PRINTBLOCK

PRINTBLOCK
=====

Synopsis:

```
PRINTBLOCK
Synopsis:
PRINTBLOCK
```

Function:

Prints the contents of the block buffer (clipboard).

1.53 FileXARexx.guide/QUIT

QUIT
=====

Synopsis:

```
QUIT FORCE/S
```

Function:

Quits FileX.

Arguments:

```
FORCE/S:
    The changes requester is suppressed.
```

See also:

CLOSE

1.54 FileXARexx.guide/REDO

REDO
=====

Synopsis:

```
REDO NUMBER/N, FILEID=FID/K/N
```

Function:

Redos a number of 'undos'.

Arguments:

```
NUMBER/N:
    Number of steps. If not given, just the last 'undo' is redone.
```

```
FILEID=FID/K/N:
    ID of a file.
```

1.55 FileXARexx.guide/REQUESTFILE

REQUESTFILE
=====

Synopsis:

```
REQUESTFILE VAR/K, STEM/K, TITLE/K, PATH/K, FILE/K, PATTERN/K =>  
FILENAME
```

Function:

Opens a file requester and returns a file name.

Arguments:

TITLE/K:
Title of the file requester.

PATH/K:
Default path.

FILE/K:
Default file.

PATTERN/K:
A pattern.

Return codes:

Falls rc = 5, if the requester was cancelled.

Sonst:
FILENAME:
Select file with path.

Example:

```
REQUESTFILE TITLE "Select a script:" PATTERN '#?.filex'
```

1.56 FileXARexx.guide/REQUESTNOTIFY

REQUESTNOTIFY
=====

Synopsis:

```
REQUESTNOTIFY PROMPT
```

Function:

Opens a simple notify requester with just one lonely gadget.

Arguments:

PROMPT:
Title of the requester.

1.57 FileXARexx.guide/REQUESTRESPONSE

REQUESTRESPONSE

=====

Synopsis:

REQUESTRESPONSE BUTTONS/K,PROMPT/K

Function:

Opens a Requester with several gadgets.

Arguments:

BUTTONS/K:

A string that defines the gadgets(see example).

PROMPT/K:

Title.

Return codes:

rc contains the number of the selected gadgets. The right most gadget has got the nuber 0. All others are enumerated from left to right beginning with 1.

Example:

```
/* Requester with four gadgets */
REQUESTRESPONSE BUTTONS '"Gadget 1|Gadget2|Gadget 3|Gadget 0"'

say '"You''ve chosen gadget number"' rc '.'
```

1.58 FileXARexx.guide/REQUESTSTRING

REQUESTSTRING

=====

Synopsis:

REQUESTSTRING VAR/K,STEM/K,PROMPT/K,DEFAULT/K => STRING

Function:

Opens a requester to enter a string.

Arguments:

PROMPT/K:

Title.

DEFAULT/K:

Default string.

Return codes:

rc = 5, if the requester was cancelled.

Else:

STRING

Entered string.

1.59 FileXARexx.guide/RX

RX

==

Synopsis:

RX VAR/K, STEM/K, CONSOLE/S, ASYNC/S, COMMAND/F => RC/N, RESULT

Function:

Executes an arexx script.

Arguments:

CONSOLE/S:

If set, a console window for input and output will be opened.

ASYNC/S:

Start the script asynchron.

COMMAND/F:

Name of the arexx script.

Return codes:

If executed synchronous.

RC/N, RESULT:

Return codes of the script.

1.60 FileXARexx.guide/SAVE

SAVE

====

Synopsis:

SAVE FILEID=FID/K/N

Function:

Saves the file. If the file does not have a name, a file requester will pop up.

Arguments:

FILEID=FID/K/N:

ID of a file.

Return codes:

rc = 5, if saving failed.

1.61 FileXARexx.guide/SAVEAS

SAVEAS
=====

Synopsis:
SAVEAS NAME, FILEID=FID/K/N

Function:
Saves the file under a new name.

Arguments:
NAME:
New name of the file. If not given, a file requester will pop up.

FILEID=FID/K/N:
ID of a file.

Return codes:
rc = 5, if saving failed.

1.62 FileXARexx.guide/SAVEBLOCK

SAVEBLOCK
=====

Synopsis:
SAVEBLOCK NAME

Function:
Saves the current contents of the block buffer (clipboard).

Arguments:
NAME:
If not given, a file requester will pop up.

Return codes:
rc = 5, if saving failed.

1.63 FileXARexx.guide/SAVECOMMANDS

SAVECOMMANDS
=====

Synopsis:
SAVECOMMANDS

Function:
Saves the arexx command list from the menu under the current name.

Return codes:
rc = 5, if saving failed.

1.64 FileXARexx.guide/SAVECOMMANDSAS

SAVECOMMANDSAS
=====

Synopsis:
SAVECOMMANDSAS NAME

Function:
Saves the arexx command list from the menu under a new name.

Arguments:
NAME:
If not given, a file requester will pop up.

Return codes:
rc = 5, if saving failed.

1.65 FileXARexx.guide/SAVESETTINGS

SAVESETTINGS
=====

Synopsis:
SAVESETTINGS

Function:
Saves the current settings under their current name.

Return codes:
rc = 5, if saving failed.

1.66 FileXARexx.guide/SAVESETTINGSAS

SAVESETTINGSAS
=====

Synopsis:
SAVESETTINGSAS NAME

Function:
Saves the current settings under a new name.

Arguments:

NAME:

If not given, a ... ehh, what do you guess? ... file requester will pop up.

Return codes:

rc = 5, if saving failed.

1.67 FileXARexx.guide/SETATTR

SETATTR

=====

Synopsis:

SETATTR OBJECT/A, FIELD, ID/K/N, FROMSTEM/K, FROMVAR/K

Function:

Set one or more variables of the program to a new value.

Arguments:

OBJECT/A:

Object name.

FIELD:

Field name.

ID/K/N:

Possible with the objects WINDOW, VIEW and FILE. If not given, the variable of the current view/window/file is set.

FROMSTEM/K:

Name of a stem variable containing the values.

FROMVAR/K:

Name of a variable containing the value.

If FIELD is not specified, a FROMSTEM-variable must be specified.

In this variable, several values of the object can be stored.

Example:

```
meinestemvariable.STRINGFILL = TRUE
meinestemvariable.FILLSTRING = 'Klaas ist stupid!'
```

```
SETATTR APPLICATION FROMSTEM meinestemvariable
```

See also:

Attributes, GETATTR

1.68 FileXARexx.guide/SETBLOCK

SETBLOCK
=====

Synopsis:
SETBLOCK BLOCK/A

Function:
Puts a number of bytes in the block buffer(clipboard).

Arguments:
BLOCK/A:
Hex string.

Example:
SETBLOCK '"0071 12feee de"'

1.69 FileXARexx.guide/SETBLOCKMARK

SETBLOCKMARK
=====

Synopsis:
SETBLOCKMARK VIEWID=VID/K/N

Function:
Switches between block mark mode and normal mode.

Arguments:
VIEWID=VID/K/N:
ID of a view.

See also:
Menu: 'Edit/Mark', Attributes: 'PROJECT/MARK', COPY, CUT, FILL

1.70 FileXARexx.guide/SETBOOKMARK

SETBOOKMARK
=====

Synopsis:
SETBOOKMARK NUMBER/N,VIEWID=VID/K/N

Function:
Sets a mark at the current cursorposition. By using GOTOBOOKMARK,
this mark can be jumped at.

Arguments:
NUMBER/N:
Number between 1 and 10, which defines the mark's number.

VIEWID=VID/K/N:
ID of a view.

See also:
GOTOBOOKMARK

1.71 FileXARexx.guide/SETBYTES

SETBYTES
=====

Synopsis:
SETBYTES POS/N/A,HEXSTRING/A,FILEID=FID/K/N

Function:
Sets a defined number of bytes at a specified position.

Arguments:
POS/N/A:
Position, the bytes shall be set to.

HEXSTRING/A:
Hex string containing the bytes.

FILEID=FID/K/N:
ID of a file.

Example:
SETBYTES 200 '"4b6c 6161 73"'

See also:
GETBYTES

1.72 FileXARexx.guide/SETSCREEN

SETSCREEN
=====

Synopsis:
SETSCREEN NAME/K,OWN/S

Function:
Definition of the screen FileX shall be opened on.

Arguments:
NAME/K:
Name of an public screen.

OWN/S:

An own screen shall be opened. A screen mode requester will pop up.

1.73 FileXARexx.guide/SHRINKVIEW

SHRINKVIEW
=====

Synopsis:

SHRINKVIEW VIEWID=VID/K/N

Function:

Shrinks a view by one line.

Arguments:

VIEWID=VID/K/N:
ID of a view.

See also:

EXPANDVIEW, GROWVIEW

1.74 FileXARexx.guide/SIZEWINDOW

SIZEWINDOW
=====

Synopsis:

SIZEWINDOW HEIGHT/N,WIDTH/N,WINDOWID=WID/K/N

Function:

The FileX window height and width may be modified.

Arguments:

HEIGHT/N:
New height.

WIDTH/N:
New width.

WINDOWID=WID/K/N:
ID of a window.

1.75 FileXARexx.guide/SPLIT

SPLIT
=====

Synopsis:

SPLIT VIEW/S,WINDOW/S

Function:

Opens a new view with the same file as the current one.

Arguments:**VIEW/S:**

The current view is divided into two new views.

WINDOW/S:

Opens a new edit window.

See also:

Menu: 'Project/(View|Window)/Split', NEW

1.76 FileXARexx.guide/UNDO

UNDO

====

Synopsis:

UNDO NUMBER/N,FILEID=FID/K/N

Function:

A defined number of changes made to the file will be undone.

Arguments:**NUMBER/N:**

Number of changes which shall be possible to be undone. If not specified, only the last change will be undone.

FILEID=FID/K/N:

ID of a file.

1.77 FileXARexx.guide/UNLOCKGUI

UNLOCKGUI

=====

Synopsis:

UNLOCKGUI

Function:

If the GUI was locked, it will be unlocked.

See also:

LOCKGUI

1.78 FileXARexx.guide/WINDOWTOBACK

WINDOWTOBACK
=====

Synopsis:

WINDOWTOBACK WINDOWID=WID/K/N

Function:

The window will be moved to the back.

Arguments:

WINDOWID=WID/K/N:
ID of a window.

1.79 FileXARexx.guide/WINDOWTOFRONT

WINDOWTOFRONT
=====

Synopsis:

WINDOWTOFRONT WINDOWID=WID/K/N

Function:

The window will be moved to the front.

Arguments:

WINDOWID=WID/K/N:
ID of a window.

1.80 FileXARexx.guide/ZIPWINDOW

ZIPWINDOW
=====

Synopsis:

ZIPWINDOW WINDOWID=WID/K/N

Function:

Same function as the zip gadget of the FileX window has got.

Arguments:

WINDOWID=WID/K/N:
ID of a window.

1.81 FileXARexx.guide/Attributes

Attributes

Some variables of FileX can be read by the option GETATTR. Probably they may also be modified through SETATTR. Both commands require at least the objectname and maybe a fieldname or an ID, too.

There are three different fieldtypes: Strings, numbers and boolean values ('TRUE' or 'FALSE').

Following are all objects and their belonging fields. For every field, its name and the type will be displayed, moreover if it can only be read. After that, a contents description follows.

APPLICATION-Objekt:

=====

"SCREEN", String, read only:

 Name of the public screen FileX is opened on.

"VERSION", String, read only:

 Versionnumber of FileX.

"REGNUMBER", Number, read only:

 Registration number. 0 statnds for the unregistered version.

"REGNAME", String, read only:

 Name of registered user.

"AREXX", String, read only:

 Name of the arexx port.

"MAXUNDOMEMSIZE", Number:

 Maximum size of the undo memory.

"MAXUNDOLEVEL", Number:

 Maximum number of undo levels.

"ALTJUMP", Number:

 Jump width in lines for Alt-CURSORUP/DOWN.

"SCROLLBORDER", Number:

 Distance in lines to the upper/lower limit from where on the window shall be scrolled.

"CLIPBOARDUNIT", Number:

 Unit that should be used for the block buffer.

"COMMANDSHELLWINDOW", String:

 Description of the command shell window.

"COMMANDWINDOW", String:

 Description of the shell window for arexx scripts.

"OVERWRITE", Bool:

 TRUE, if no requester before saving an already existing file shall

appear.

"USEASL", Bool:

TRUE, if the Asl library and the Intuition library should be used instead of the Reqtools library.

"PRIORITY", Number:

Taskpriority of FileX.

"SEARCHSTRING", String:

Search string.

"REPLACESTRING", String:

Replace string.

"STRINGSEARCH", Bool:

TRUE for string search, FALSE for hex search.

"CASESENSITIVSEARCH", Bool:

TRUE for casesensitiv searching.

"WILDSEARCH", Bool:

TRUE to use the question mark('?') as a wildcard.

"FILLSTRING", String:

Fill string.

"STRINGFILL", Bool:

TRUE for stringfill, FALSE for hex fill.

"WINDOWS"-Objekt:

=====

1..n

ID of a window.

"COUNT", Number, read only:

Number of IDs, i.e. the number of opened edit windows.

"WINDOW"-Objekt:

=====

"LEFT", Number, read only:

Window offset to the left border.

"TOP", Number, read only:

Window offset to the right border.

"WIDTH", Number, read only:

Width of the window.

"HEIGHT", Number, read only:

Height of the window.

"TITLE", String, read only:

Current title of the window.

"MIN_WIDTH", Number, read only:
Minimum width of the window.

"MIN_HEIGHT", Number, read only:
Minimum height of the window.

"MAX_WIDTH", Number, read only:
Maximum width of the window.

"MAX_HEIGHT", Number, read only:
Minimum height of the window.

"VIEWCOUNT", Number, read only:
Number of views.

"VIEWS", String, read only:
IDs of all views in this window. Separated by a space.

"FILES"-Objekt:
=====

1..n
ID of a file.

"COUNT", Number, read only:
Number of IDs, i.e. the number of opened files.

"FILE"-Objekt:
=====

"FULLNAME", String:
Full file name with path.

"PATH", String:
Path of the file name.

"NAME", String:
Name without path.

"CHANGES", Number, read only:
Number of changes.

"FILELEN", Number, read only:
Length of the file.

"MEMPOS", Number, read only:
Start address of the file in the memory. If the length of the file
is changed, the start address might change, too.

"VIEWCOUNT", Number, read only:
Number of views showing the file.

"VIEWS", String, read only:
IDs of all views showing the file. Separated by a space.

"VIEW"-Objekt:
=====

"LINES", Number, read only:
Number of lines.

"BYTESPERLINE", Number, read only:
Number of bytes per line.

"DISPLAYTYP", Number:
Current display type.

"DISPLAY", Number:
Defines in which form the file shall be displayed: 1 = only
hexddisplay, 2 = only Asciiidisplay, 3 = Hex-ASCII mixed display.

"DISPLAYSPACES", Number:
Defines how many Bytes shall be grouped to a block in the
hexdisplay: 3 = no blocks, 0 = one byte blocks, 1 = word blocks, 2
= long word blocks.

"CURSORPOS", Number, read only:
Current cursor position.

"MARKPOS", Number, read only:
Position of the mark's beginning.

"EDITASCII", Bool:
TRUE, when the active cursor is in the ascii area.

"MARK", Bool, read only:
TRUE, if a block is being marked at the moment.

"FILE", Number, read only:
ID of the displayed file.

"WINDOW", Number, read only:
ID of the belonging edit window.
