

Deutsch/FileXARexx

COLLABORATORS

	<i>TITLE :</i> Deutsch/FileXARexx		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		September 19, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Deutsch/FileXARexx	1
1.1	Deutsch/FileXARexx.guide	1
1.2	FileXARexx.guide/Einführung	1
1.3	FileXARexx.guide/Kommandoübersicht	2
1.4	FileXARexx.guide/ACTIVATE	6
1.5	FileXARexx.guide/ACTIVATEVIEW	7
1.6	FileXARexx.guide/ACTIVATEWINDOW	7
1.7	FileXARexx.guide/ADJUSTWINDOWSIZE	8
1.8	FileXARexx.guide/APPEND	8
1.9	FileXARexx.guide/ASSIGNCOMMAND	8
1.10	FileXARexx.guide/CHANGEWINDOW	9
1.11	FileXARexx.guide/CLEAR	10
1.12	FileXARexx.guide/CLOSE	10
1.13	FileXARexx.guide/COLUMN	11
1.14	FileXARexx.guide/COPY	11
1.15	FileXARexx.guide/CURSOR	11
1.16	FileXARexx.guide/CUT	12
1.17	FileXARexx.guide/DEACTIVATE	12
1.18	FileXARexx.guide/EXECUTECOMMAND	13
1.19	FileXARexx.guide/EXPANDVIEW	13
1.20	FileXARexx.guide/FAULT	14
1.21	FileXARexx.guide/FILL	14
1.22	FileXARexx.guide/FIND	15
1.23	FileXARexx.guide/FINDCHANGE	16
1.24	FileXARexx.guide/FINDNEXT	17
1.25	FileXARexx.guide/FONT	17
1.26	FileXARexx.guide/GETATTR	18
1.27	FileXARexx.guide/GETBLOCK	19
1.28	FileXARexx.guide/GETBYTES	19
1.29	FileXARexx.guide/GOTOBOOKMARK	20

1.30	FileXARexx.guide/GOTOBYTE	20
1.31	FileXARexx.guide/GOTOCOLUMN	21
1.32	FileXARexx.guide/GOTOLINE	21
1.33	FileXARexx.guide/GRABMEMORY	21
1.34	FileXARexx.guide/GROWVIEW	22
1.35	FileXARexx.guide/HELP	22
1.36	FileXARexx.guide/INSERT	23
1.37	FileXARexx.guide/LINE	23
1.38	FileXARexx.guide/LOADBLOCK	24
1.39	FileXARexx.guide/LOADCOMMANDS	24
1.40	FileXARexx.guide/LOADDISPLAYTYP	25
1.41	FileXARexx.guide/LOADSETTINGS	25
1.42	FileXARexx.guide/LOCKGUI	25
1.43	FileXARexx.guide/MOVEWINDOW	26
1.44	FileXARexx.guide/NEW	26
1.45	FileXARexx.guide/NEXT	27
1.46	FileXARexx.guide/NOP	27
1.47	FileXARexx.guide/OPEN	27
1.48	FileXARexx.guide/PASTE	28
1.49	FileXARexx.guide/POSITION	28
1.50	FileXARexx.guide/PREVIOUS	29
1.51	FileXARexx.guide/PRINT	29
1.52	FileXARexx.guide/PRINTBLOCK	30
1.53	FileXARexx.guide/QUIT	30
1.54	FileXARexx.guide/REDO	30
1.55	FileXARexx.guide/REQUESTFILE	31
1.56	FileXARexx.guide/REQUESTNOTIFY	32
1.57	FileXARexx.guide/REQUESTRESPONSE	32
1.58	FileXARexx.guide/REQUESTSTRING	33
1.59	FileXARexx.guide/RX	33
1.60	FileXARexx.guide/SAVE	34
1.61	FileXARexx.guide/SAVEAS	34
1.62	FileXARexx.guide/SAVEBLOCK	34
1.63	FileXARexx.guide/SAVECOMMANDS	35
1.64	FileXARexx.guide/SAVECOMMANDSAS	35
1.65	FileXARexx.guide/SAVESETTINGS	35
1.66	FileXARexx.guide/SAVESETTINGSAS	36
1.67	FileXARexx.guide/SETATTR	36
1.68	FileXARexx.guide/SETBLOCK	37

1.69	FileXARexx.guide/SETBLOCKMARK	37
1.70	FileXARexx.guide/SETBOOKMARK	38
1.71	FileXARexx.guide/SETBYTES	38
1.72	FileXARexx.guide/SETSCREEN	39
1.73	FileXARexx.guide/SHRINKVIEW	39
1.74	FileXARexx.guide/SIZEWINDOW	40
1.75	FileXARexx.guide/SPLIT	40
1.76	FileXARexx.guide/UNDO	41
1.77	FileXARexx.guide/UNLOCKGUI	41
1.78	FileXARexx.guide/WINDOWTOBACK	41
1.79	FileXARexx.guide/WINDOWTOFRONT	42
1.80	FileXARexx.guide/ZIPWINDOW	42
1.81	FileXARexx.guide/Attribute	42

Chapter 1

Deutsch/FileXARexx

1.1 Deutsch/FileXARexx.guide

\$VER: FileXARexxDokumentation 2.0 (6.5.94)

FileX - Ein Binäreditor

© Copyright 1993-1994 bei Klaas Hermanns

Shareware, alle Rechte vorbehalten.

Einführung

Kommandoübersicht

Attribute

1.2 FileXARexx.guide/Einführung

Einführung

FileX hat eine weitgehend an die Richtlinien aus Commodores Amiga User Interface Style Guide angelehnte ARexxschnittstelle.

Falls nicht über das Schlüsselwort PORTNAME beim Start ein anderer Name angegeben wird, erhält der Port den Namen 'FILEX'. Sollte unter diesem Namen bereits ein Port existieren, wird an den Namen eine Zahl angehängt und diese solange erhöht bis ein noch nicht existierender Portname gefunden wird ('NAME' -> 'NAME.1' -> 'NAME.2').

Die Argumente der Befehle werden AmigaDOS üblich ausgewertet. Falls die verschiedenen Typen nicht bekannt sein sollten, hier ein kurze Auflistung:

<arg>
Eine einfache Zeichenkette. Es ist nicht notwendig den Namen des Arguments anzugeben.

<arg>/S
Schalter. Durch Angabe des Arguments wird er gesetzt.

<arg>/K
Schlüsselwort. Das Argument muß mit dem Argumentnamen angegeben werden ('<arg> = wert' oder '<arg> wert').

<arg>/N
Nummer. Das Argument muß eine Dezimalzahl sein.

<arg>/T
Wechselschalter. Durch Angabe des Arguments ändert ein Wert seinen Zustand.

<arg>/A
Das Argument muß angegeben werden.

<arg>/F
Der Rest der Zeile wird als Argument angenommen, auch wenn dort noch Schlüsselwörter vorkommen.

<arg>/M
Mehrere Werte sind möglich.

Bei allen Kommandos, die einen Wert zurückgeben, ist es möglich durch Angabe einer Variablen mittels des Arguments VAR, eine Variable auf das Ergebnis setzen zu lassen. Gleiches gilt für das Argument STEM und Stemvariablen.

Alle Kommandos, denen eine ID übergeben werden kann, liefern rc = 5, falls es sich bei der ID um eine ungültige ID handelt. Falls keine ID angegeben wird, wird für das Kommando die aktuelle Ansicht, das aktuelle Fenster oder die aktuelle Datei benutzt.

1.3 FileXARexx.guide/Kommandoübersicht

Kommandoübersicht (alphabetische geordnet)

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
=====

Aufbau:
 ACTIVATE

Funktion:
 Aktiviert FileX aus dem Iconify.

Beispiel:
 DEACTIVATE /* Entspricht Menu: 'Project/Iconify' */

 /* Jetzt liegt ein Icon auf der Workbench und FileX schläft */

 ACTIVATE /* Fenster wird wieder geöffnet */

Siehe auch:
 DEACTIVATE

1.5 FileXARexx.guide/ACTIVATEVIEW

ACTIVATEVIEW
=====

Aufbau:
 ACTIVATEVIEW VIEWID=VID/K/N

Funktion:
 Aktiviert eine Ansicht.

Argumente:
 VIEWID=VID/K/N:
 ID der Ansicht, die aktiviert werden soll.

1.6 FileXARexx.guide/ACTIVATEWINDOW

ACTIVATEWINDOW
=====

Aufbau:
 ACTIVATEWINDOW WINDOWID=WID/K/N

Funktion:
 Aktiviert ein Edierfenster.

Argumente:
 WINDOWID=WID/K/N:
 ID eines Fensters, das aktiviert werden soll.

1.7 FileXARexx.guide/ADJUSTWINDOWSIZE

ADJUSTWINDOWSIZE
=====

Aufbau:

ADJUSTWINDOWSIZE WINDOWID=WID/K/N

Funktion:

Paßt die Größe eines Fensters, an die Größe der Darstellungsboxen an.

Argumente:

WINDOWID=WID/K/N:
ID eines Fensters.

Siehe auch:

Menu: 'Settings/Adjust window size'

1.8 FileXARexx.guide/APPEND

APPEND
=====

Aufbau:

APPEND VIEWID=VID/K/N

Funktion:

Der aktuelle Block wird an das Dateiende angehängt.

Argumente:

VIEWID=VID/K/N:
ID einer Ansicht, zu der die Datei gehört.

Siehe auch:

Menu: 'Edit/Append'

1.9 FileXARexx.guide/ASSIGNCOMMAND

ASSIGNCOMMAND
=====

Aufbau:

ASSIGNCOMMAND NUMBER/K/N, NAME/K

Funktion:

Setzt eines der ARexxkommandos im Menu 'ARexx'.

Argumente:

NUMBER/K/N:

Nummer zwischen 0 und 10, die die Position des Kommandos bestimmt. Falls nicht angegeben, erscheint ein Requester zur Eingabe der Nummer.

NAME/K:

Filename des Kommandos. Falls nicht angegeben, erscheint ein Filerequester zur Auswahl eines Files.

Beispiel:

```
ASSIGNCOMMAND 0 'dh0:meinearexxbefehle/filexbefehl'
```

Siehe auch:

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

1.10 FileXARexx.guide/CHANGEWINDOW

CHANGEWINDOW

=====

Aufbau:

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

Funktion:

Verändert die Größe und/oder Lage des Edierfensters.

Argumente:

LEFTEDGE/K/N:
Neuer linker Abstand des Fensters.

TOPEDGE/K/N:
Neuer oberer Abstand des Fensters.

HEIGHT/K/N:
Neue Höhe des Fensters.

WIDTH/K/N:
Neue Breite des Fensters.

WINDOWID=WID/K/N
ID eines Windows, das geändert werden soll.

Falls ein Wert nicht angegeben, bleibt er unverändert.

Beispiel:

```
/* Fenster auf maximale Größe bringen */  
CHANGEWINDOW LEFTEDGE 0 TOPEDGE 0 HEIGHT 4000 WIDTH 4000
```

Siehe auch:

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

1.11 FileXARexx.guide/CLEAR

CLEAR
=====

Aufbau:

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

Funktion:

Löscht eine Anzeige.

Argumente:

FORCE/S:

Die Sicherheitsabfrage wird unterdrückt.

VIEWID=VID/K/N

ID einer Ansicht, die gelöscht werden soll.

Rückgabewerte:

rc = 5, falls die Sicherheitsabfrage abgebrochen wurde.

Siehe auch:

Menu: 'Project/New'

1.12 FileXARexx.guide/CLOSE

CLOSE
=====

Aufbau:

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

Funktion:

Schließt eine Ansicht bzw. ein ganzes Edierfenster.

Argumente:

VIEW/S:

Falls gesetzt, wird die ID als Ansicht ID aufgefasst und es wird versucht eine Ansicht zu schließen.

WINDOW/S:

Falls gesetzt, wird die ID als Fenster ID aufgefasst und es wird versucht ein Edierfenster zu schließen.

FORCE/S:

Falls gesetzt, wird der Sicherheitsrequester unterbunden, der über verlorengelassene Änderungen informiert.

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

ID der Ansicht bzw. des Fensters, das geschlossen werden soll.

Siehe auch:

QUIT

1.13 FileXARexx.guide/COLUMN

COLUMN
=====

Aufbau:

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

Funktion:

Verschiebt den Cursor nach rechts oder links.

Argumente:

DELTA/N/A:

Anzahl der Schritte, um die der Cursor verschoben werden soll. Positive Zahlen verschieben den Cursor nach rechts, negative nach links.

VIEWID=VID/K/N

ID einer Ansicht, in der der Cursor verschoben werden soll.

Siehe auch:

GOTOBYTE, GOTOCOLUMN, GOTOLINE, LINE

1.14 FileXARexx.guide/COPY

COPY
=====

Aufbau:

COPY VIEWID=VID/K/N

Funktion:

Kopiert den aktuell markierten Block in den Blockspeicher.

Argumente:

VIEWID=VID/K/N

ID einer Ansicht, aus der der Block kopiert werden soll.

Siehe auch:

Menu: 'Edit/Copy', SETBLOCKMARK

1.15 FileXARexx.guide/CURSOR

CURSOR
=====

Aufbau:

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

Funktion:

Verschiebt den Cursor.

Argumente:**UP/S:**

Den Cursor um eins nach oben verschieben.

DOWN/S:

Den Cursor um eins nach unten verschieben.

LEFT/S:

Den Cursor um eins nach links verschieben.

RIGHT/S:

Den Cursor um eins nach rechts verschieben.

VIEWID=VID/K/N:

ID einer Ansicht, in der der Cursor verschoben werden soll.

UP und DOWN sowie LEFT und RIGHT schliessen sich gegenseitig aus.

Siehe auch:

GOTOBYTE, LINE, COLUMN

1.16 FileXARexx.guide/CUT

CUT

====

Aufbau:

CUT VIEWID=VID/K/N

Funktion:

Kopiert den aktuell markierten Block in den Blockspeicher und schneidet ihn dabei aus der Datei aus.

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

ID einer Ansicht, aus der der Block kopiert werden soll

Siehe auch:

Menu: `Edit/Cut`, SETBLOCKMARK

1.17 FileXARexx.guide/DEACTIVATE

DEACTIVATE

=====

Aufbau:

DEACTIVATE

Funktion:

Bringt FileX in den Iconify-Modus. Das Fenster wird geschlossen und ein Icon auf die Workbench erzeugt.

ACHTUNG!: Alle ARexxbefehle, die eine Ausgabe in einem Fenster machen oder diese in einer anderen Art verändern, können während des Iconify-Modus zum Absturz führen!

Siehe auch:

ACTIVATE

1.18 FileXARexx.guide/EXECUTECOMMAND

EXECUTECOMMAND

=====

Aufbau:

EXECUTECOMMAND NUMBER/K/N, NAME/K

Funktion:

Startet ein ARexxkommando.

Argumente:

NUMBER/K/N:

Nummer zwischen 0 und 9 eines mit ASSIGNCOMMAND installierten Kommandos.

NAME/K:

Name eines ARexxscript. Beide Argumente schließen sich aus.

Siehe auch:

ASSIGNCOMMAND

1.19 FileXARexx.guide/EXPANDVIEW

EXPANDVIEW

=====

Aufbau:

EXPANDVIEW VIEWID=VID/K/N

Funktion:

Vergrößert die aktuelle Ansicht auf ihre maximale Größe. Alle anderen Ansichten im Fenster sind nach dem Kommando nur noch eine Zeile hoch.

Argumente:

VIEWID=VID/K/N:

ID einer Ansicht, die auf maximale Größe gebracht werden soll.

Siehe auch:

SHRINKVIEW, GROWVIEW

1.20 FileXARexx.guide/FAULT

FAULT

=====

Aufbau:

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

Funktion:

Ermittelt den zur Dosfehlernummer gehörenden Fehlertext.

Argumente:

NUMBER/N/A:

Dosfehlernummer.

Rückgabewerte:

DESCRIPTION:

Der zugehörige Text.

Beispiel:

FAULT 114

```
/* Result hat bei lokalisierter Workbbench den Wert: */
```

```
/* `: Falsches Namensmuster' */
```

1.21 FileXARexx.guide/FILL

FILL

=====

Aufbau:

FILL VIEWID=VID/K/N

Funktion:

Füllt den aktuell markierten Bereich mit dem aktuellem Füllstring.

Argumente:

VIEWID=VID/K/N:

ID einer Ansicht, in der der aktuell markierte Block gefüllt werden soll.

Beispiel:

```
/* Falls gerade ein Block markiert ist, */
/* die Markierung aufheben */

GETATTR VIEW MARK
if result=TRUE then SETBLOCKMARK

/* Füllstring setzten auf 'Füll füllherum!' */

SETATTR APPLICATION FILLSTRING FROMVAR ' "Füll füllherum!" '

/* Cursor auf Start des Füllbereiches setzen */
GOTOBYTE 10

/* Blockmarke setzen */
SETBLOCKMARK

/* Cursor verschieben */
GOTOBYTE 100

/* Bereich füllen */
FILL

/* Blockmarkierung aufheben */
SETBLOCKMARK
```

Siehe auch:

Menu: 'Edit/Fill', SETBLOCKMARK, SETATTR

1.22 FileXARexx.guide/FIND

FIND

====

Aufbau:

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

Funktion:

Sucht in der aktuellen Ansicht nach einer Zeichenkette oder einer Hexzahlenfolge und positioniert bei Erfolg den Cursor an den Anfang dieser. Die Suche beginnt an der aktuellen Cursorposition.

Argumente:

PROMPT/S:

Öffnet den Suchrequester. BACKWARDS und QUIET werden dabei ignoriert.

HEX/S:

Bei den Suchtext handelt es sich um eine Hexzahlenfolge.

BACKWARDS/S:

Rückwärts suchen

QUIET/S:

Keinen Requester bei Mißerfolg öffnen.

TEXT/F:

Der Suchtext oder die Suchhexzahlenfolge.

VIEWID=VID/K/N:

ID einer Ansicht, in der gesucht werden soll.

Rückgabewerte:

rc = 5, falls die Zeichenkette nicht gefunden werden konnte.

Beispiel:

```
/* Sucht nach '$VER:' ohne bei Mißerfolg einen */
/* Requester zu öffnen */
FIND QUIET TEXT '$VER:'
```

Siehe auch:

Attribute: 'APPLICATION/#?SEARCH', FINDNEXT, FINDCHANGE, Menu:
'Search&Replace/#?'

1.23 FileXARexx.guide/FINDCHANGE

FINDCHANGE

=====

Aufbau:

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

Funktion:

Sucht in der Datei nach einer Zeichenkette oder einer Hexzahlenfolge und ersetzt diese durch eine andere (ggf. erscheint vorher ein Requester, der fragt, ob die Stelle wirklich ersetzt werden soll). Die Suche beginnt an der aktuellen Cursorposition.

Argumente:

PROMPT/S:

Öffnet den Suchenundersetzenrequester. ALL, BACKWARDS und QUIET werden dabei ignoriert.

HEX/S:

Bei dem Suchtext handelt es sich um eine Hexzahlenfolge.

ALL/S:

Alle gefunden Stellen werden ohne einen Requester ersetzt.

BACKWARDS/S:

Rückwärts suchen.

QUIET/S:

Keinen Requester bei Mißerfolg öffnen.

FIND:

Der Suchtext oder die Suchhexzahlenfolge.

CHANGE:

Der Ersetzentext oder die Ersetzen- hexzahlenfolge.

VIEWID=VID/K/N:

ID einer Ansicht, in der gesucht werden soll.

Rückgabewerte:

rc = 5, falls die Zeichenkette nicht gefunden werden konnte.

Beispiel:

```
/* Ersetzt alle 'Claas' durch 'Klaas' */  
FINDCHANGE QUIET ALL 'Claas' 'Klaas'  
FINDCHANGE BACKWARDS QUIET ALL 'Claas' 'Klaas'
```

Siehe auch:

FINDNEXT, FIND, Menu: 'Search&Replac/#?'

1.24 FileXARexx.guide/FINDNEXT

FINDNEXT

=====

Aufbau:

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

Funktion:

Setzt die letzte Suche fort, d.h. entweder Suchen oder Suchen_und_Ersetzen.

Argumente:

BACKWARDS/S:

Rückwärts weitersuchen.

QUIET/S:

Keine Requester öffnen.

VIEWID=VID/K/N:

ID einer Ansicht, in der gesucht werden soll.

Rückgabewerte:

rc = 5, falls Suche fehlschlägt.

Siehe auch:

FIND, FINDCHANGE

1.25 FileXARexx.guide/FONT

FONT
=====

Aufbau:

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

Funktion:

Setzt den von einem Edierfenster benutzten Zeichensatz.

Argumente:

NAME/A:

Name des Fonts.

SIZE/N/A:

Größe des Fonts.

WINDOWID=WID/K/N:

ID eines Edierfensters, dessen Zeichensatz geändert werden soll.

Beispiel:

FONT 'horror.font' 8

1.26 FileXARexx.guide/GETATTR

GETATTR
=====

Aufbau:

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

Funktion:

Ermittelt den Wert einer oder mehrerer im Programm verwendeten Variablen.

Argumente:

OBJECT/A:

Name des Objekts.

FIELD:

Name des Felds.

ID/K/N:

Möglich bei den Objekten WINDOW, VIEW und FILE. Falls nicht angegeben, werden bei diesen Objekten Informationen des aktuellen Edierfensters, der aktuellen Ansicht oder der aktuellen Datei ermittelt.

Falls FIELD nicht angegeben wird, muß eine STEM-Variable angegeben werden, in der dann alle Werte des Objekts abgelegt werden.

Beispiel:

```
GETATTR WINDOW TITLE
```

```
/* Result enthält jetzt den Titel des Windows */  
/* Nicht sehr interessant, ich weiß :-) */
```

Siehe auch:

Attribute, SETATTR

1.27 FileXARexx.guide/GETBLOCK

```
GETBLOCK
```

```
=====
```

Aufbau:

```
GETBLOCK VAR/K,STEM/K => BLOCK
```

Funktion:

Ermittelt den Inhalt des Blockspeichers. Dieser entspricht dem Inhalt des Clipboards.

Rückgabewerte:

BLOCK:
Enthält den Block als Hexzeichenfolge.

Siehe auch:

SETBLOCK

1.28 FileXARexx.guide/GETBYTES

```
GETBYTES
```

```
=====
```

Aufbau:

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

Funktion:

Ermittelt den Wert von einer bestimmten Anzahl Bytes an einer Stelle der Datei.

Argumente:

POS/N/A:
Position der zur ermittelnden Bytes.

NUMBER/N/A:
Anzahl der Bytes (max:32767).

Rückgabewerte:

HEXSTRING:
Hexzahlenfolge der Bytes.

FILEID=FID/K/N:

ID einer Datei, aus der die Bytefolge bestimmt werden soll.

Siehe auch:

SETBYTES

1.29 FileXARexx.guide/GOTOBOOKMARK

GOTOBOOKMARK

=====

Aufbau:

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

Funktion:

Setzt den Cursor an eine vorher mit SETBOOKMARK bestimmte Stelle.

Argumente:

NUMBER/N/A:

Nummer zw. 1 und 10 der Stelle.

VIEWID=VID/K/N:

ID der Ansicht, in der der Cursor bewegt werden soll.

Siehe auch:

SETBOOKMARK

1.30 FileXARexx.guide/GOTOBYTE

GOTOBYTE

=====

Aufbau:

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

Funktion:

Setzt den Cursor an eine bestimmte Stelle.

Argumente:

POS/N/A:

Abstand in Bytes zum Dateianfang.

VIEWID=VID/K/N:

ID einer Ansicht, in der der Cursor bewegt werden soll.

Siehe auch:

COLUMN, GOTOCOLUMN, GOTOLINE, LINE

1.31 FileXARexx.guide/GOTOCOLUMN

GOTOCOLUMN
=====

Aufbau:

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

Funktion:

Setzt den Cursor in eine bestimmte Spalte.

Argumente:

COLUMN/N/A:

Spalte zwischen 1 und 16.

VIEWID=VID/K/N:

ID einer Ansicht, in der der Cursor bewegt werden soll.

Siehe auch:

COLUMN, GOTOBYTE, GOTOLINE, LINE

1.32 FileXARexx.guide/GOTOLINE

GOTOLINE
=====

Aufbau:

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

Funktion:

Setzt den Cursor in eine bestimmte Zeile.

Argumente:

LINE/N/A:

Zeilennummer.

VIEWID=VID/K/N:

ID einer Ansicht, in der der Cursor bewegt werden soll.

Siehe auch:

COLUMN, GOTOBYTE, GOTOCOLUMN, LINE

1.33 FileXARexx.guide/GRABMEMORY

GRABMEMORY
=====

Aufbau:

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

Funktion:

Stellt einen Speicherbereich statt eines Files dar.

Argumente:

START/N/A:

Anfang des Speicherbereichs.

END/N/A:

Ende des Speicherbereichs.

FORCE/S:

Die Sicherheitsabfrage bei Änderungen wird unterbunden.

Rückgabewerte:

rc = 5, falls der Befehl scheiterte (Sicherheitsabfrage abgebrochen).

Beispiel:

```
GRABMEMORY 200000 280000 FORCE
```

Siehe auch:

Menu: 'Project/Grab memory'

1.34 FileXARexx.guide/GROWVIEW

GROWVIEW

=====

Aufbau:

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

Funktion:

Eine Ansicht wird um eine Zeile vergrößert, falls es möglich ist.

Argumente:

VIEWID=VID/K/N:

ID einer Ansicht, die vergrößert werden soll.

Siehe auch:

SHRINKVIEW, EXPANDVIEW

1.35 FileXARexx.guide/HELP

HELP

=====

Aufbau:

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

Funktion:

Ermittelt entweder eine List aller Befehle der ARexxschnittstelle von FileX oder den Aufbau eines speziellen Befehls.

Argumente:**COMMAND:**

Name des Kommandos, von dem der Aufbau ermittelt werden soll.

Falls COMMAND nicht angegeben wird, erhält man eine Kommandoliste.

Rückgabewerte:

Falls COMMAND angegeben:

COMMANDDESC:

Aufbau des Kommandos. Sonst:

COMMANDLIST/M:

Kommandozahl und eine Liste aller Kommandos.

1.36 FileXARexx.guide/INSERT

INSERT

=====

Aufbau:

INSERT VIEWID=VID/K/N

Funktion:

Fügt den Block aus dem Blockspeicher an der aktuellen Cursorposition in die Datei ein.

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

ID einer Ansicht, in der der Block eingefügt werden soll.

Siehe auch:

Menu: 'Edit/Insert', SETBLOCKMARK

1.37 FileXARexx.guide/LINE

LINE

=====

Aufbau:

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

Funktion:

Verschiebt den Cursor um eine bestimmte Anzahl Zeilen.

Argumente:**LINE/N/A:**

Anzahl der Zeilen. Positiv nach unten.

VIEWID=VID/K/N:

ID einer Ansicht, in der der Cursor bewegt werden soll.

Siehe auch:

COLUMN, GOTobyte, GOTOCOLUMN, GOTOLINE

1.38 FileXARexx.guide/LOADBLOCK

LOADBLOCK

=====

Aufbau:

LOADBLOCK NAME

Funktion:

Läd einen Block in den Blockspeicher.

Argumente:

NAME:

Dateiname des Blocks. Falls nicht angegeben, öffnet sich ein Filerequester zur Auswahl.

Rückgabewerte:

rc = 5, falls das Laden scheiterte.

Siehe auch:

SAVEBLOCK

1.39 FileXARexx.guide/LOADCOMMANDS

LOADCOMMANDS

=====

Aufbau:

LOADCOMMANDS NAME

Funktion:

Läd ARExxkommandos ein.

Argumente:

NAME:

Dateiname der Kommandos. Falls nicht angegeben, öffnet sich ein Filerequester zur Auswahl.

Rückgabewerte:

rc = 5, falls das Laden scheiterte.

Siehe auch:

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

1.40 FileXARexx.guide/LOADDISPLAYTYP

LOADDISPLAYTYP
=====

Aufbau:

LOADDISPLAYTYP NAME

Funktion:

Läd den frei bestimmbaren Displaytyp ein.

Argumente:

NAME:

Dateiname des Displaytyp. Falls nicht angegeben, öffnet sich ein Filerequester zur Auswahl.

Rückgabewerte:

rc = 5, falls das Laden scheiterte.

Siehe auch:

Menu: 'Settings/Display/#?'

1.41 FileXARexx.guide/LOADSETTINGS

LOADSETTINGS
=====

Aufbau:

LOADSETTINGS NAME

Funktion:

Läd eine Einstellungsdatei ein.

Argumente:

NAME:

Dateiname der Einstellungsdatei. Falls nicht angegeben, öffnet sich ein Filerequester zur Auswahl.

Rückgabewerte:

rc = 5, falls das Laden scheiterte.

Siehe auch:

Menu: 'Settings/Load Settings...'

1.42 FileXARexx.guide/LOCKGUI

LOCKGUI
=====

Aufbau:
 LOCKGUI

Funktion:
 Sperrt alle Edierfenster für Benutzereingaben. Nur die Größe und Position der Fenster kann noch geändert werden.

Rückgabewerte:
 rc = 5, falls die Fenster nicht gesperrt werden können. Dies ist der Fall, wenn noch ein Requester offen ist.

Siehe auch:
 LOCKGUI

1.43 FileXARexx.guide/MOVEWINDOW

MOVEWINDOW
=====

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

Funktion:
 Verändert die Lage des Edierfensters.

Argumente:
 LEFTEDGE/N:
 Neuer linker Abstand des Fensters.

 TOPEDGE/N:
 Neuer oberer Abstand des Fensters.

 WINDOWID=WID/K/N:
 ID eines Edierfensters, das bewegt werden soll.

1.44 FileXARexx.guide/NEW

NEW
===

Aufbau:
 NEW VIEW/S, WINDOW/S

Funktion:
 Eine neue Ansicht wird geöffnet.

Argumente:

VIEW/S:

Die aktuelle Ansicht wird in zwei Hälften geteilt.

WINDOW/S:

Es wird ein neues Fenster geöffnet.

Siehe auch:

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

1.45 FileXARexx.guide/NEXT

NEXT

====

Aufbau:

NEXT VIEW/S,WINDOW/S

Funktion:

Es wird die nächste Ansicht oder das nächste Edierfenster aktiviert.

Argumente:

VIEW/S:

Die nächste Ansicht im aktuellen Edierfenster wird aktiviert.

WINDOW/S:

Das nächste Edierfenster wird aktiviert.

Siehe auch:

PREVIOUS

1.46 FileXARexx.guide/NOP

NOP

====

Aufbau:

NOP

Funktion:

Diese Kommando macht nichts. Wirklich rein gar nichts:-)

1.47 FileXARexx.guide/OPEN

OPEN

====

Aufbau:

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

Funktion:

Öffnet eine Datei.

Argumente:

FILENAME:

Name der Datei. Falls nicht angegeben, öffnet sich ein
Filerequester zur Auswahl.

FORCE/S:

Die Sicherheitsabfrage wird unterdrückt.

VIEWID=VID/K/N:

ID einer Ansicht, in der eine neue Datei geladen werden soll.

Rückgabewerte:

rc = 5, falls das Laden scheiterte.

Siehe auch:

CLEAR, SAVE, SAVEAS

1.48 FileXARexx.guide/PASTE

PASTE

=====

Aufbau:

PASTE VIEWID=VID/K/N

Funktion:

Schreibt den Block aus dem Blockspeicher an der aktuellen
Cursorposition in die Datei.

Argumente:

VIEWID=VID/K/N:

ID der Ansicht, in die der Block geschrieben werden soll.

Siehe auch:

Menu: 'Edit/Paste', SETBLOCKMARK

1.49 FileXARexx.guide/POSITION

POSITION

=====

Aufbau:

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

Funktion:

Positioniert den Cursor.

Argumente:**SOF/S:**

Cursor an den Anfang der Datei.

EOF/S:

Cursor an das Ende der Datei.

SOL/S:

Cursor an den Anfang der Zeile.

EOL/S:

Cursor an das Ende der Zeile.

VIEWID=VID/K/N:

ID einer Ansicht, in der der Cursor bewegt werden soll.

SOF und EOF sowie SOL und EOL schließen sich gegenseitig aus.

1.50 FileXARexx.guide/PREVIOUS

PREVIOUS

=====

Aufbau:

PREVIOUS VIEW/S,WINDOW/S

Funktion:

Es wird die vorherige Ansicht oder das vorherige Edierfenster aktiviert.

Argumente:**VIEW/S:**

Die vorherige Ansicht im aktuellen Edierfenster wird aktiviert.

WINDOW/S:

Das vorherige Edierfenster wird aktiviert.

Siehe auch:

NEXT

1.51 FileXARexx.guide/PRINT

PRINT
=====

Aufbau:
PRINT FILEID=FID/K/N

Funktion:
Druckt eine vollständige Datei aus.

Argumente:
ID einer Datei, die ausgedruckt werden soll.

1.52 FileXARexx.guide/PRINTBLOCK

PRINTBLOCK
=====

Aufbau:
PRINTBLOCK
Aufbau:
PRINTBLOCK

Funktion:
Druck den Block aus dem Blockspeicher.

1.53 FileXARexx.guide/QUIT

QUIT
=====

Aufbau:
QUIT FORCE/S

Funktion:
Beendet das Programm.

Argumente:
FORCE/S:
Eine Sicherheitsabfrage wird unterdrückt. unterdrückt.

Siehe auch:
CLOSE

1.54 FileXARexx.guide/REDO

REDO
=====

Aufbau:

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

Funktion:

Eine bestimmte Anzahl 'Undos' wird rückgängig gemacht.

Argumente:

NUMBER/N:

Anzahl der 'Undos', die rückgängig gemacht werden sollen.
Falls nicht angegeben, wird nur das letzte 'Undo'
rückgängig gemacht.

FILEID=FID/K/N:

ID einer Datei, in der die 'Undos' rückgängig gemacht
werden sollen.

1.55 FileXARexx.guide/REQUESTFILE

REQUESTFILE
=====

Aufbau:

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

Funktion:

Öffnet einen Filerequester und gibt ggf. einen ausgewählten
Dateinamen zurück.

Argumente:

TITLE/K:

Titel des Filerequesters.

PATH/K:

Voreingestellter Pfadname.

FILE/K:

Voreingestellter Filename.

PATTERN/K:

Muster.

Rückgabewerte:

Falls rc = 5, wurde der Filerequester abgebrochen.

Sonst:

FILENAME:

Ausgewählter Filename mit Pfad.

Beispiel:

```
REQUESTFILE TITLE 'Bitte ein Script auswählen:'' PATTERN  
'#?.filex'
```

1.56 FileXARexx.guide/REQUESTNOTIFY

```
REQUESTNOTIFY  
=====
```

Aufbau:

```
REQUESTNOTIFY PROMPT
```

Funktion:

Öffnet eine einfachen Mitteilungsrequester mit einem Gadget.

Argumente:

```
PROMPT:  
Titeltext.
```

1.57 FileXARexx.guide/REQUESTRESPONSE

```
REQUESTRESPONSE  
=====
```

Aufbau:

```
REQUESTRESPONSE BUTTONS/K,PROMPT/K
```

Funktion:

Öffnet einen Requester mit mehreren Gadgets zur Auswahl und liefert die Nummer des angewählten Gadgets.

Argumente:

```
BUTTONS/K:  
Eine Zeichenkette, die den Aufbau der Gadgets bestimmt (siehe  
Beispiel).
```

```
PROMPT/K:  
Titeltext.
```

Rückgabewerte:

rc enthält die Nummer des ausgewählten Gadgets. Das Gadget, das ganz rechts liegt, hat die Nummer 0. Alle anderen werden von links nach rechts mit 1 beginnend durchnummeriert.

Beispiel:

```
/* Requster mit 4 Gadgets öffnen */  
REQUESTRESPONSE BUTTONS '"Gadget 1|Gadget2|Gadget 3|Gadget 0"'  
  
say 'Du hast Gadget Nummer' rc 'ausgewählt.'
```

1.58 FileXARexx.guide/REQUESTSTRING

REQUESTSTRING

=====

Aufbau:

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

Funktion:

Öffnet einen Requester zur Eingabe einer Zeichenkette.

Argumente:

PROMPT/K:
Titeltext.

DEFAULT/K:
Voreingestellte Zeichenkette.

Rückgabewerte:

rc = 5, falls der Requester abgebrochen worden ist.

Sonst:
STRING
Eingegebene Zeichenkette.

1.59 FileXARexx.guide/RX

RX

==

Aufbau:

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

Funktion:

Startet ein ARexxscript.

Argumente:

CONSOLE/S:
Falls angegeben, wird ein Fenster für ein und ausgabe geöffnet.

ASYNC/S:
Das Script soll asynchron ausgeführt werden.

COMMAND/F:
Der Name des Kommandos.

Rückgabewerte:

Falls synchron ausgeführt:
RC/N,RESULT:
Rückgabewerte des Scripts.

1.60 FileXARexx.guide/SAVE

SAVE
=====

Aufbau:

```
SAVE FILEID=FID/K/N
```

Funktion:

Speichert die Datei unter dem aktuellen Namen ab. Falls die Datei noch keinen Namen hatte, erscheint ein Filerequester.

Funktion:

Argumente:

FILEID=FID/K/N:

ID einer Datei, die gespeichert werden soll.

Rückgabewerte:

rc = 5, falls die Datei nicht abgespeichert werden konnte.

1.61 FileXARexx.guide/SAVEAS

SAVEAS
=====

Aufbau:

```
SAVEAS NAME,FILEID=FID/K/N
```

Funktion:

Speichert die Datei unter einem neuen Namen ab.

Argumente:

NAME:

Name, unter dem die Datei abgespeichert werden soll. Falls nicht angegeben, erscheint ein Filerequester.

FILEID=FID/K/N:

ID einer Datei, die gespeichert werden soll.

Rückgabewerte:

rc = 5, falls die Datei nicht abgespeichert werden konnte.

1.62 FileXARexx.guide/SAVEBLOCK

SAVEBLOCK
=====

Aufbau:

```
SAVEBLOCK NAME
```

Funktion:

Speichert den Block aus dem Blockspeicher ab.

Argumente:**NAME:**

Name, unter dem der Block abgespeichert werden soll. Falls nicht angegeben, erscheint ein Filerequester.

Rückgabewerte:

rc = 5, falls der Block nicht abgespeichert werden konnte.

1.63 FileXARexx.guide/SAVECOMMANDS

SAVECOMMANDS

=====

Aufbau:

SAVECOMMANDS

Funktion:

Speichert die im Menu installierten ARexxkommandos unter dem aktuellen Namen ab.

Rückgabewerte:

rc = 5, falls die Kommandos nicht abgespeichert werden konnten.

1.64 FileXARexx.guide/SAVECOMMANDSAS

SAVECOMMANDSAS

=====

Aufbau:

SAVECOMMANDSAS NAME

Funktion:

Speichert die im Menu installierten ARexxkommandos unter einem neuen Namen ab.

Argumente:**NAME:**

Dateiname der Kommandos. Falls nicht angegeben, öffnet sich ein Filerequester.

Rückgabewerte:

rc = 5, falls die Kommandos nicht abgespeichert werden konnten.

1.65 FileXARexx.guide/SAVESETTINGS

SAVESETTINGS
=====

Aufbau:
SAVESETTINGS

Funktion:
Speichert die aktuellen Einstellungen unter dem aktuellen Namen ab.

Rückgabewerte:
rc = 5, falls die Einstellungen nicht abgespeichert werden konnten.

1.66 FileXARexx.guide/SAVESETTINGSAS

SAVESETTINGSAS
=====

Aufbau:
SAVESETTINGSAS NAME

Funktion:
Speichert die aktuellen Einstellungen unter einem neuen Namen ab.

Argumente:
NAME:
Dateiname der Einstellungen. Falls nicht angegeben, öffnet sich ein Filerequester.

Rückgabewerte:
rc = 5, falls die Einstellungen nicht abgespeichert werden konnten.

1.67 FileXARexx.guide/SETATTR

SETATTR
=====

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

Funktion:
Setzt den Wert einer oder mehrerer im Programm verwendeten Variablen.

Argumente:
OBJECT/A:
Name des Objekts.

FIELD:

Name des Felds.

ID/K/N:

Möglich bei den Objekten WINDOW, VIEW und FILE. Falls nicht angegeben, werden bei diesen Objekten Informationen des aktuellen Edierfensters, der aktuellen Ansicht oder der aktuellen Datei gesetzt.

FROMSTEM/K:

Name einer STEM-Variablen, die die zu setzenden Werte enthält.

FROMVAR/K:

Name einer Variablen, die einen zu setzenden Wert enthält.

Falls FIELD nicht angegeben wird, muß eine FROMSTEM-Variable angegeben werden, in der dann mehrere Werte des Objekts enthalten sein können.

Beispiel:

```
meinestemvariable.STRINGFILL = TRUE
meinestemvariable.FILLSTRING = 'Klaas ist blöd.'

SETATTR APPLICATION FROMSTEM meinestemvariable
```

Siehe auch:

Attribute, GETATTR

1.68 FileXARexx.guide/SETBLOCK

SETBLOCK
=====

Aufbau:

```
SETBLOCK BLOCK/A
```

Funktion:

Setzt eine Bytefolge im Blockspeicher. Dieser entspricht dem Clipboard.

Argumente:

BLOCK/A:
Hexzahlenfolge.

Beispiel:

```
SETBLOCK '"0071 12feee de"'
```

1.69 FileXARexx.guide/SETBLOCKMARK

SETBLOCKMARK
=====

Aufbau:

SETBLOCKMARK VIEWID=VID/K/N

Funktion:

Wechselt zwischen dem Block-Markieren-Modus und dem normalen Modus.

Argumente:

VIEWID=VID/K/N:

ID einer Ansicht, in der der Modus gewechselt werden soll.

Siehe auch:

Menu: 'Edit/Mark', Attribute: 'PROJECT/MARK', COPY, CUT, FILL

1.70 FileXARexx.guide/SETBOOKMARK

SETBOOKMARK
=====

Aufbau:

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

Funktion:

Setzen einer Marke an der aktuellen Position im der Datei, die über GOTOBOOKMARK wieder angesprungen werden kann.

Argumente:

NUMBER/N:

Nummer zwischen 1 und 10, die die Nummer der Marke bestimmt.

VIEWID=VID/K/N:

ID einer Ansicht, von der die Cursorposition gemerkt werden soll.

Siehe auch:

GOTOBOOKMARK

1.71 FileXARexx.guide/SETBYTES

SETBYTES
=====

Aufbau:

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

Funktion:

Setzen einer bestimmten Anzahl Bytes an einer bestimmten Position.

Argumente:

POS/N/A:

Position, an der die Bytes gesetzt werden sollen.

HEXSTRING/A:

Hexzahlenfolge der Bytes.

FILEID=FID/K/N:

ID einer Datei, in der die Bytefolge gesetzt werden soll.

Beispiel:

```
SETBYTES          200 '"4b6c 6161 73"'
```

Siehe auch:

GETBYTES

1.72 FileXARexx.guide/SETSCREEN

SETSCREEN

=====

Aufbau:

```
SETSCREEN NAME/K,OWN/S
```

Funktion:

Bestimmung des Screens auf den das FileXfenster geöffnet sein soll.

Argumente:

NAME/K:

Name eines Publicscreens.

OWN/S:

Es soll ein eigener Screen geöffnet werden. Ein Requester läßt den User Breite, Höhe usw. einstellen.

1.73 FileXARexx.guide/SHRINKVIEW

SHRINKVIEW

=====

Aufbau:

```
SHRINKVIEW VIEWID=VID/K/N
```

Funktion:

Eine Ansicht wird um eine Zeile verkleinert, falls es möglich ist.

Argumente:

VIEWID=VID/K/N:

ID einer Ansicht, die verkleinert werden soll.

Siehe auch:

EXPANDVIEW, GROWVIEW

1.74 FileXARexx.guide/SIZEWINDOW

SIZEWINDOW

=====

Aufbau:

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

Funktion:

Die Höhe und die Breite des Edierfensters werden geändert.

Argumente:

HEIGHT/N:

Neue Höhe des Fensters.

WIDTH/N:

Neue Breite des Fensters.

WINDOWID=WID/K/N:

ID eines Edierfensters, dessen Größe verändert werden soll.

1.75 FileXARexx.guide/SPLIT

SPLIT

=====

Aufbau:

SPLIT VIEW/S,WINDOW/S

Funktion:

Eine neue Ansicht wird geöffnet, die die gleiche Datei anzeigt wie die aktuelle Ansicht.

Argumente:

VIEW/S:

Die aktuelle Ansicht wird in zwei Hälften geteilt.

WINDOW/S:

Es wird ein neues Fenster geöffnet.

Siehe auch:

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

1.76 FileXARexx.guide/UNDO

UNDO
=====

Aufbau:

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

Funktion:

Eine bestimmte Anzahl Veränderungen der Datei werden rückgängig gemacht.

Argumente:

NUMBER/N:

Anzahl der Veränderungen, die rückgängig gemacht werden sollen. Falls nicht angegeben, wird nur die letzte Änderung rückgängig gemacht.

FILEID=FID/K/N:

ID einer Datei, in der die Veränderung(en) rückgängig gemacht werden soll.

1.77 FileXARexx.guide/UNLOCKGUI

UNLOCKGUI
=====

Aufbau:

UNLOCKGUI

Funktion:

Der Befehl LOCKGUI wird wieder aufgehoben. Der Benutzer kann wieder Eingaben machen.

Siehe auch:

LOCKGUI

1.78 FileXARexx.guide/WINDOWTOBACK

WINDOWTOBACK
=====

Aufbau:

WINDOWTOBACK WINDOWID=WID/K/N

Funktion:

Das Edierfenster wird in den Hintergrund gebracht.

Argumente:

WINDOWID=WID/K/N:

ID eines Edierfensters, das in den Hintergrund gebracht werden soll.

1.79 FileXARexx.guide/WINDOWTOFRONT

WINDOWTOFRONT
=====

Aufbau:

WINDOWTOFRONT WINDOWID=WID/K/N

Funktion:

Das Edierfenster wird in den Vordergrund gebracht.

Argumente:

WINDOWID=WID/K/N:

ID eines Edierfensters, das in den Vordergrund gebracht werden soll.

1.80 FileXARexx.guide/ZIPWINDOW

ZIPWINDOW
=====

Aufbau:

ZIPWINDOW WINDOWID=WID/K/N

Funktion:

Gleiche Funktion wie das Zipgadget eines Fensters. Das Edierfenster wird gezippt.

Argumente:

WINDOWID=WID/K/N:

ID eines Edierfensters, das gezippt werden soll.

1.81 FileXARexx.guide/Attribute

Attribute

Einige Variablen des Programms können durch den Befehl GETATTR abgerufen und eventuell auch durch SETATTR verändert werden. Beide Befehle erwarten mindestens einen Objekt- und gegebenenfalls auch einen Feldnamen oder eine ID.

Es gibt drei verschiedene Feldtypen: Texte, Zahlen und boolsche Wahrheitswerte ('TRUE' (wahr) oder 'FALSE' (nicht wahr)).

Nachfolgend alle Objekte mit zugehörigen Feldern. Für jedes Feld wird der Name und der Typ genannt. Ggf. noch, ob man es nur lesen kann. Darauf folgt eine Beschreibung des Inhalts.

APPLICATION-Objekt:

=====

"SCREEN", Text, nur lesbar:

Name des öffentlichen Bildschirms auf dem Filex sein Fenster geöffnet hat.

"VERSION", Text, nur lesbar:

Versionsnummer als Zeichenkette.

"REGNUMBER", Zahl, nur lesbar:

Registriernummer. 0 für unregistriert.

"REGNAME", Text, nur lesbar:

Name des Benutzers, auf den diese Version registriert ist.

"AREXX", Text, nur lesbar:

Name des ARexxports.

"MAXUNDOMEMSIZE", Zahl:

Maximale Größe des Undospeichers.

"MAXUNDOLEVEL", Zahl:

Maximale Undozahl.

"ALTJUMP", Zahl:

Sprungweite in Zeilen von Alt-CURSORUP/DOWN.

"SCROLLBORDER", Zahl:

Abstand in Zeilen zur oberen/unteren Begrenzung, ab der eine Ansicht gescrollt wird.

"CLIPBOARDUNIT", Zahl:

Nummer des Clipboards für den Blockspeicher.

"COMMANDSHELLWINDOW", Text:

Beschreibung des Commandshellwindows.

"COMMANDWINDOW", Text:

Beschreibung des Ein-/Ausgabefenster für ARexxscripts.

"OVERWRITE", Bool:

TRUE, falls keine Abfrage vor dem Speichern einer bereits existierenden Datei kommen soll.

"USEASL", Bool:

TRUE, falls die Asllibrary und Intuitionlibrary für die Requester gebraucht werden sollen. Andernfalls wird die Reqtoolslibrary gebraucht.

"PRIORITY", Zahl:

Taskpriorität von FileX.

"SEARCHSTRING", Text:
Suchzeichenkette oder Hexzahlenfolge.

"REPLACESTRING", Text:
Ersatzzeichenkette oder Hexzahlenfolge.

"STRINGSEARCH", Bool:
TRUE, falls nach einer Zeichenkette gesucht werden soll. FALSE, um nach eine Hexzahlenfolge zu suchen.

"CASESENSITIVSEARCH", Bool:
Falls TRUE, wird bei der Textsuche zwischen Groß- und Kleinschreibung unterschieden.

"WILDSEARCH", Bool:
Falls TRUE, wird das Fragezeichen('?') bei der Textsuche als Platzhalter für ein beliebiges Zeichen verstanden.

"FILLSTRING", Text:
Zeichenkette oder Hexzahlenfolge für die Füllfunktion.

"STRINGFILL", Bool:
TRUE, falls der Fillstring als Zeichenkette aufgefasst werden soll.

"WINDOWS"-Objekt:
=====

1..n
ID eines Fensters.

"COUNT", Zahl, nur lesbar:
Anzahl der IDs, d.h. gleichzeitig Anzahl der offenen Fenster.

"WINDOW"-Objekt:
=====

"LEFT", Zahl, nur lesbar:
Linker Randabstand des Fensters.

"TOP", Zahl, nur lesbar:
Rechter Randabstand des Fensters.

"WIDTH", Zahl, nur lesbar:
Breite des Fensters.

"HEIGHT", Zahl, nur lesbar:
Höhe des Fensters.

"TITLE", Text, nur lesbar:
Aktueller Titel des Fensters.

"MIN_WIDTH", Zahl, nur lesbar:
Minimale Breite des Fensters.

"MIN_HEIGHT", Zahl, nur lesbar:
Minimale Höhe des Fensters.

"MAX_WIDTH", Zahl, nur lesbar:
Maximale Breite des Fensters.

"MAX_HEIGHT", Zahl, nur lesbar:
Maximale Höhe des Fensters.

"VIEWCOUNT", Zahl, nur lesbar:
Anzahl der Ansichten.

"VIEWS", Text, nur lesbar:
IDs aller Ansichten, die im Edierfenster darstellt werden. Durch
ein Leerzeichen voneinander getrennt.

"FILES"-Objekt:
=====

1..n
ID einer Datei.

"COUNT", Zahl, nur lesbar:
Anzahl der IDs, d.h. gleichzeitig Anzahl der offenen Dateien.

"FILE"-Objekt:
=====

"FULLNAME", Text:
Voller Name mit gesamtem Pfad.

"PATH", Text:
Pfad der Datei.

"NAME", Text:
Name der Datei.

"CHANGES", Zahl, nur lesbar:
Anzahl der Änderungen in der Datei.

"FILELEN", Zahl, nur lesbar:
Länge der Datei.

"MEMPOS", Zahl, nur lesbar:
Die Anfangsadresse der Datei im Speicher. Durch Änderungen an der
Datei, die die Länge verändern, kann sich auch die
Anfangsadresse ändern.

"VIEWCOUNT", Zahl, nur lesbar:
Anzahl der Ansichten, die die Datei darstellen.

"VIEWS", Text, nur lesbar:
IDs aller Ansichten, die die Datei darstellen. Durch ein
Leerzeichen voneinander getrennt.

"VIEW"-Objekt:
=====

"LINES", Zahl, nur lesbar:

Anzahl der Zeilen.

"BYTESPERLINE", Zahl, nur lesbar:
Anzahl der Bytes in einer Zeile.

"DISPLAYTYP", Zahl:
Aktuell eingestellter Displaytyp.

"DISPLAY", Zahl:
Bestimmt die Aufteilung der Ansicht: 1 = nur Hexdarstellung, 2 =
nur Asciiarstellung, 3 = Hex-ASCII gemischte Darstellung.

"DISPLAYSPACES", Zahl:
Bestimmt die Größe der Blöcke bei der Hexdarstellung: 3 = keine
Zwischenräume in der Hexdarstellung, 0 = Nach jedem Byte ein
Leerzeichen, 1 = alle zwei Bytes ein Leerzeichen, 2 = alle vier
Bytes ein Leerzeichen.

"CURSORPOS", Zahl, nur lesbar:
Aktuelle Cursorposition.

"MARKPOS", Zahl, nur lesbar:
Position des Markierungsanfangs.

"EDITASCII", Bool:
TRUE, falls der aktive Cursor im Asciibereich steht.

"MARK", Bool, nur lesbar:
TRUE, falls gerade ein Block markiert wird.

"FILE", Zahl, nur lesbar:
ID der Datei, die dargestellt wird.

"WINDOW", Zahl, nur lesbar:
ID des Edierfensters, in dem sich die Anzeige befindet.
