

Deutsch/FileXARexx

COLLABORATORS

	TITLE : Deutsch/FileXARexx		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		January 8, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

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

1.30 FileXARexx.guide/HELP	17
1.31 FileXARexx.guide/INSERT	18
1.32 FileXARexx.guide/LINE	18
1.33 FileXARexx.guide/LOADBLOCK	19
1.34 FileXARexx.guide/LOADBLOCKCLIP	19
1.35 FileXARexx.guide/LOADCOMMANDS	19
1.36 FileXARexx.guide/LOADDISPLAYTYP	20
1.37 FileXARexx.guide/LOADSETTINGS	20
1.38 FileXARexx.guide/MOVEWINDOW	21
1.39 FileXARexx.guide/NOP	21
1.40 FileXARexx.guide/OPEN	21
1.41 FileXARexx.guide/PASTE	22
1.42 FileXARexx.guide/POSITION	22
1.43 FileXARexx.guide/PRINT	23
1.44 FileXARexx.guide/PRINTBLOCK	23
1.45 FileXARexx.guide/QUIT	23
1.46 FileXARexx.guide/REDO	23
1.47 FileXARexx.guide/REQUESTFILE	24
1.48 FileXARexx.guide/REQUESTNOTIFY	24
1.49 FileXARexx.guide/REQUESTRESPONSE	25
1.50 FileXARexx.guide/REQUESTSTRING	25
1.51 FileXARexx.guide/RX	26
1.52 FileXARexx.guide/SAVE	26
1.53 FileXARexx.guide/SAVEAS	27
1.54 FileXARexx.guide/SAVEBLOCK	27
1.55 FileXARexx.guide/SAVEBLOCKCLIP	27
1.56 FileXARexx.guide/SAVECOMMANDS	28
1.57 FileXARexx.guide/SAVECOMMANDSAS	28
1.58 FileXARexx.guide/SAVESETTINGS	29
1.59 FileXARexx.guide/SAVESETTINGSAS	29
1.60 FileXARexx.guide/SETATTR	29
1.61 FileXARexx.guide/SETBLOCK	30
1.62 FileXARexx.guide/SETBLOCKMARK	30
1.63 FileXARexx.guide/SETBOOKMARK	31
1.64 FileXARexx.guide/SETBYTES	31
1.65 FileXARexx.guide/SETSCREEN	32
1.66 FileXARexx.guide/SIZEWINDOW	32
1.67 FileXARexx.guide/UNDO	32
1.68 FileXARexx.guide/WINDOWTOBACK	33
1.69 FileXARexx.guide/WINDOWTOFRONT	33
1.70 FileXARexx.guide/ZIPWINDOW	33
1.71 FileXARexx.guide/Attribute	33

Chapter 1

Deutsch/FileXARexx

1.1 Deutsch/FileXARexx.guide

```
$VER: FileXARexxDokumentation 1.1 (16.10.93)
```

```
FileX - Ein Binäreditor
```

```
© Copyright 1993 bei Klaas Hermanns
```

```
Shareware, alle Rechte vorbehalten.
```

```
Generelles  
Kommandoübersicht  
Attribute
```

1.2 FileXARexx.guide/Generelles

```
Generelles  
*****
```

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 angehängen 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.

1.3 FileXARexx.guide/Kommandoübersicht

Kommandoübersicht (alphabetische geordnet)

Name:	Argumente => Rückgabewerte:

ACTIVATE	
ACTIVATEWINDOW	
APPEND	
ASSIGNCOMMAND	NUMBER/N/K, NAME/K
CHANGEWINDOW	LEFTEDGE/N/K, TOPEDGE/N/K, HEIGHT/N/K, WIDTH/N/K
CLEAR	FORCE/S
COLUMN	DELTA/N/A
COPY	
CURSOR	UP/S, DOWN/S, LEFT/S, RIGHT/S
CUT	
DEACTIVATE	
EXECUTECOMMAND	NUMBER/N/K, NAME/K
FAULT	VAR/K, STEM/K, NUMBER/N/A => DESCRIPTION
FILL	
FIND	PROMPT/S, HEX/S, BACKWARDS/S, QUIET/S, TEXT/F
FINDCHANGE	PROMPT/S, HEX/S, ALL/S, BACKWARDS/S, QUIET/S, FIND, CHANGE

FINDNEXT	BACKWARDS/S, QUIET/S
FONT	NAME/A, SIZE/A/N
GETATTR	OBJECT/A, FIELD, STEM/K, VAR/K
GETBLOCK	VAR/K, STEM/K => BLOCK
GETBYTES	VAR/K, STEM/K, POS/N/A, NUMBER/N/A => HEXSTRING
GOTOBOOKMARK	NUMBER/N
GOTOBYTE	POS/N/A
GOTOCOLUMN	COLUMN/N/A
GOTOLINE	LINE/N/A
GRABMEMORY	START/A/N, END/A/N, FORCE/S
HELP	VAR/K, STEM/K, COMMAND => COMMANDDESC, COMMANDLIST/M
INSERT	
LINE	DELTA/N/A
LOADBLOCK	NAME
LOADBLOCKCLIP	
LOADCOMMANDS	NAME
LOADDISPLAYTYP	NAME
LOADSETTINGS	NAME
MOVEWINDOW	LEFTEDGE/N, TOPEDGE/N
NOP	
OPEN	FILENAME, FORCE/S
PASTE	
POSITION	SOF/S, EOF/S, SOL/S, EOL/S
PRINT	
PRINTBLOCK	
QUIT	FORCE/S
REDO	NUMBER/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	
SAVEAS	NAME
SAVEBLOCK	NAME
SAVEBLOCKCLIP	
SAVECOMMANDS	
SAVECOMMANDSAS	NAME
SAVESETTINGS	
SAVESETTINGSAS	NAME
SETATTR	OBJECT/A, FIELD, FROMSTEM/K, FROMVAR/K
SETBLOCK	BLOCK/A
SETBLOCKMARK	
SETBOOKMARK	NUMBER/N
SETBYTES	POS/N/A, HEXSTRING/A
SETSCREEN	NAME/K, OWN/S
SIZEWINDOW	HEIGHT/N, WIDTH/N
UNDO	NUMBER/N
WINDOWTOBACK	
WINDOWTOFRONT	
ZIPWINDOW	

Name: Argumente => Rückgabewerte:

ACTIVATE
ACTIVATEWINDOW
APPEND

ASSIGNCOMMAND	NUMBER/N/K, NAME/K
CHANGEWINDOW	LEFTEDGE/N/K, TOPEDGE/N/K, HEIGHT/N/K, WIDTH/N/K
CLEAR	FORCE/S
COLUMN	DELTA/N/A
COPY	
CURSOR	UP/S, DOWN/S, LEFT/S, RIGHT/S
CUT	
DEACTIVATE	
EXECUTECOMMAND	NUMBER/N/K, NAME/K
FAULT	VAR/K, STEM/K, NUMBER/N/A => DESCRIPTION
FILL	
FIND	PROMPT/S, HEX/S, BACKWARDS/S, QUIET/S, TEXT/F
FINDCHANGE	PROMPT/S, HEX/S, ALL/S, BACKWARDS/S, QUIET/S, FIND, CHANGE
FINDNEXT	BACKWARDS/S, QUIET/S
FONT	NAME/A, SIZE/A/N
GETATTR	OBJECT/A, FIELD, STEM/K, VAR/K
GETBLOCK	VAR/K, STEM/K => BLOCK
GETBYTES	VAR/K, STEM/K, POS/N/A, NUMBER/N/A => HEXSTRING
GOTOBOOKMARK	NUMBER/N
GOTOBYTE	POS/N/A
GOTOCOLUMN	COLUMN/N/A
GOTOLINE	LINE/N/A
GRABMEMORY	START/A/N, END/A/N, FORCE/S
HELP	VAR/K, STEM/K, COMMAND => COMMANDDESC, COMMANDLIST/M
INSERT	
LINE	DELTA/N/A
LOADBLOCK	NAME
LOADBLOCKCLIP	
LOADCOMMANDS	NAME
LOADDISPLAYTYP	NAME
LOADSETTINGS	NAME
MOVEWINDOW	LEFTEDGE/N, TOPEDGE/N
NOP	
OPEN	FILENAME, FORCE/S
PASTE	
POSITION	SOF/S, EOF/S, SOL/S, EOL/S
PRINT	
PRINTBLOCK	
QUIT	FORCE/S
REDO	NUMBER/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	
SAVEAS	NAME
SAVEBLOCK	NAME
SAVEBLOCKCLIP	
SAVECOMMANDS	
SAVECOMMANDSAS	NAME
SAVESETTINGS	
SAVESETTINGSAS	NAME
SETATTR	OBJECT/A, FIELD, FROMSTEM/K, FROMVAR/K
SETBLOCK	BLOCK/A
SETBLOCKMARK	
SETBOOKMARK	NUMBER/N

SETBYTES	POS/N/A, HEXSTRING/A
SETSCREEN	NAME/K, OWN/S
SIZEWINDOW	HEIGHT/N, WIDTH/N
UNDO	NUMBER/N
WINDOWTOBACK	
WINDOWTOFRONT	
ZIPWINDOW	

Befehlsbeschreibungen

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/ACTIVATEWINDOW

ACTIVATEWINDOW
=====

Aufbau:

ACTIVATEWINDOW

Funktion:

Aktiviert das FileXfenster.

1.6 FileXARexx.guide/APPEND

APPEND

=====

Aufbau:

APPEND

Funktion:

Der aktuelle Block wird an das Dateiende angehängen.

Siehe auch:

Menu: 'Edit/Append'

1.7 FileXARexx.guide/ASSIGNCOMMAND

ASSIGNCOMMAND

=====

Aufbau:

ASSIGNCOMMAND NUMBER/N/K,NAME/K

Funktion:

Setzt eines der ARexxkommandos im Menu 'ARexx'.

Argumente:

NUMBER/N/K:

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

1.8 FileXARexx.guide/CHANGEWINDOW

CHANGEWINDOW

=====

Aufbau:

CHANGEWINDOW LEFTEDGE/N/K, TOPEDGE/N/K, HEIGHT/N/K, WIDTH/N/K

Funktion:

Verändert die Größe und/oder Lage des FileXfensters.

Argumente:

LEFTEDGE/N/K:

Neuer linker Abstand des Fensters.

TOPEDGE/N/K:

Neuer oberer Abstand des Fensters.

HEIGHT/N/K:

Neue Höhe des Fensters.

WIDTH/N/K:

Neue Breite des Fensters.

Falls ein Wert nicht angegeben, bleibt er unverändert.

Beispiel:

```
/* Fenster auf maximale Größe bringen */  
CHANGEWINDOW LEFTEDGE 0 TOPEDGE 0 HEIGHT 900
```

Siehe auch:

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

1.9 FileXARexx.guide/CLEAR

CLEAR

=====

Aufbau:

CLEAR FORCE/S

Funktion:

Entfernt das aktuelle File aus dem Speicher.

Argumente:

FORCE/S:

Der Requester 'You made x changes' wird unterdrückt.

Rückgabewerte:

rc = 5, falls das File nicht entfernt wurde.

Siehe auch:

Menu: 'Project/New'

1.10 FileXARexx.guide/COLUMN

COLUMN

=====

Aufbau:

COLUMN DELTA/N/A

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 und negative nach links.

Siehe auch:

GOTOBYTE, GOTOCOLUMN, GOTOLINE, LINE

1.11 FileXARexx.guide/COPY

COPY

====

Aufbau:

COPY

Funktion:

Kopiert den aktuell markierten Block in den Blockspeicher.

Siehe auch:

Menu: 'Edit/Copy', SETBLOCKMARK

1.12 FileXARexx.guide/CURSOR

CURSOR

=====

Aufbau:

CURSOR UP/S, DOWN/S, LEFT/S, RIGHT/S

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.

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

Siehe auch:

GOTOBYTE, LINE, COLUMN

1.13 FileXARexx.guide/CUT

CUT

===

Aufbau:

CUT

Funktion:

Kopiert den aktuell markierten Block in den Blockspeicher und schneidet ihn dabei aus dem File.

Siehe auch:

Menu: 'Edit/Cut', SETBLOCKMARK

1.14 FileXARexx.guide/DEACTIVATE

DEACTIVATE

=====

Aufbau:

DEACTIVATE

Funktion:

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

Siehe auch:

ACTIVATE

1.15 FileXARexx.guide/EXECUTECOMMAND

EXECUTECOMMAND

=====

Aufbau:

EXECUTECOMMAND NUMBER/N/K, NAME/K

Funktion:

Startet ein ARexxkommando.

Argumente:

NUMBER/N/K:

Nummer zwischen 0 und 9 eines mit ASSIGNCOMMAND installierten Kommandos.

NAME/K:

Name eines ARexxscript.

Beide Argumente schließen sich aus.

Siehe auch:

ASSIGNCOMMAND

1.16 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' */
```

1.17 FileXARexx.guide/FILL

FILL

=====

Aufbau:

FILL

Funktion:

Füllt den aktuell markierten Bereich mit dem aktuellem

Füllstring.

Beispiel:

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

GETATTR PROJECT MARK
if result=TRUE then SETBLOCKMARK

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

SETATTR PROJECT 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

1.18 FileXARexx.guide/FIND

FIND

====

Aufbau:

FIND PROMPT/S,HEX/S,BACKWARDS/S,QUIET/S,TEXT/F

Funktion:

Sucht in der Datei 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.

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

1.19 FileXARexx.guide/FINDCHANGE

FINDCHANGE
=====

Aufbau:
FINDCHANGE PROMPT/S,HEX/S,ALL/S,BACKWARDS/S,QUIET/S,FIND,CHANGE

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 den 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.

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&Replace/#?'

1.20 FileXARexx.guide/FINDNEXT

FINDNEXT

=====

Aufbau:

FINDNEXT BACKWARDS/S, QUIET/S

Funktion:

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

Argumente:

BACKWARDS/S:

Rückwärts weitersuchen.

QUIET/S:

Keine Requester öffnen.

Rückgabewerte:

rc = 5, falls Suche fehlschlägt.

Siehe auch:

FIND, FINDCHANGE

1.21 FileXARexx.guide/FONT

FONT

=====

Aufbau:

FONT NAME/A, SIZE/N/A

Funktion:

Setzt den verwendeten Font.

Argumente:

NAME/A:
Name des Fonts.

SIZE/N/A:
Größe des Fonts.

Beispiel:

FONT 'horror.font' 8

1.22 FileXARexx.guide/GETATTR

GETATTR
=====

Aufbau:

GETATTR OBJECT/A, FIELD, 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.

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 */  
/* z.B.: 'File: ENV:Sys/overscan.prefs (70 Bytes)' */
```

Siehe auch:

Attribute, SETATTR

1.23 FileXARexx.guide/GETBLOCK

GETBLOCK
=====

Aufbau:

GETBLOCK VAR/K, STEM/K => BLOCK

Funktion:

Ermittelt den Inhalt des Blockspeichers.

Rückgabewerte:**BLOCK:**

Enthält den Block als Hexzeichenfolge.

Siehe auch:

SETBLOCK

1.24 FileXARexx.guide/GETBYTES

GETBYTES

=====

Aufbau:

GETBYTES VAR/K,STEM/K,POS/N/A,NUMBER/N/A => 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.

Siehe auch:

SETBYTES

1.25 FileXARexx.guide/GOTOBOOKMARK

GOTOBOOKMARK

=====

Aufbau:

GOTOBOOKMARK NUMBER/N/A

Funktion:

Setzt den Cursor an eine vorher mit SETBOOKMARK bestimmte Stelle.

Argumente:**NUMBER/N/A:**

Nummer zw. 1 und 10 der Stelle.

Siehe auch:
SETBOOKMARK

1.26 FileXARexx.guide/GOTOBYTE

GOTOBYTE
=====

Aufbau:
GOTOBYTE POS/N/A

Funktion:
Setzt den Cursor an eine bestimmte Stelle.

Argumente:
POS/N/A:
Position.

Siehe auch:
COLUMN, GOTOCOLUMN, GOTOLINE, LINE

1.27 FileXARexx.guide/GOTOCOLUMN

GOTOCOLUMN
=====

Aufbau:
GOTOCOLUMN COLUMN/N/A

Funktion:
Setzt den Cursor in eine bestimmte Spalte.

Argumente:
COLUMN/N/A:
Spalte zwischen 1 und 16.

Siehe auch:
COLUMN, GOTOBYTE, GOTOLINE, LINE

1.28 FileXARexx.guide/GOTOLINE

GOTOLINE
=====

Aufbau:
GOTOLINE LINE/N/A

Funktion:

Setzt den Cursor in eine bestimmte Zeile.

Argumente:

LINE/N/A:

Zeilennummer.

Siehe auch:

COLUMN, GOTOBYTE, GOTOCOLUMN, LINE

1.29 FileXARexx.guide/GRABMEMORY

GRABMEMORY

=====

Aufbau:

GRABMEMORY START/N/A,END/N/A,FORCE/S

Funktion:

Stellt einen Speicherbereich statt eines Files dar.

Argumente:

START/N/A:

Anfang des Speicherbereichs.

END/N/A:

Ende des Speicherbereichs.

FORCE/S:

Der Requester 'You made x changes' wird unterdrückt.

Rückgabewerte:

rc = 5, falls der Befehl scheiterte (Requester gecancelt).

Beispiel:

GRABMEMORY 200000 280000 FORCE

Siehe auch:

Menu: 'Project/Grab memory'

1.30 FileXARexx.guide/HELP

HELP

=====

Aufbau:

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

Funktion:

Ermittelt entweder eine Übersicht über alle unterstützten Befehle der ARexxschnittstelle oder den Aufbau eines 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.

1.31 FileXARexx.guide/INSERT

INSERT

=====

Aufbau:

INSERT

Funktion:

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

Siehe auch:

Menu: 'Edit/Insert', SETBLOCKMARK

1.32 FileXARexx.guide/LINE

LINE

=====

Aufbau:

LINE DELTA/N/A

Funktion:

Verschiebt den Cursor um eine bestimmte Anzahl Zeilen.

Argumente:

LINE/N/A:

Anzahl der Zeilen. Positiv nach unten.

Siehe auch:

COLUMN, GOTOBYTE, GOTOCOLUMN, GOTOLINE

1.33 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

1.34 FileXARexx.guide/LOADBLOCKCLIP

LOADBLOCKCLIP
=====

Aufbau:

LOADBLOCKCLIP

Funktion:

Kopiert den aktuellen Inhalt des Clipboards in den Blockspeicher.

1.35 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

1.36 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/#?'

1.37 FileXARexx.guide/LOADSETTINGS

LOADSETTINGS
=====

Aufbau:

LOADSETTINGS NAME

Funktion:

Läd eine Einstellungsdatei ein.

Argumente:

NAME:

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

Rückgabewerte:

rc = 5, falls das Laden scheiterte.

Siehe auch:

Menu: 'Settings/Load Settings...'

1.38 FileXARexx.guide/MOVEWINDOW

MOVEWINDOW
=====

Aufbau:

MOVEWINDOW LEFTEDGE/N, TOPEDGE/N

Funktion:

Verändert die Lage des FileXfensters.

Argumente:

LEFTEDGE/N:

Neuer linker Abstand des Fensters.

TOPEDGE/N:

Neuer oberer Abstand des Fensters.

Falls ein Wert nicht angegeben, bleibt er unverändert.

Siehe auch:

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

1.39 FileXARexx.guide/NOP

NOP
===

Aufbau:

NOP

Funktion:

Diese Kommando macht (leider?) mal nichts, gar nichts:-)

1.40 FileXARexx.guide/OPEN

OPEN
====

Aufbau:

OPEN FILENAME, FORCE/S

Funktion:

Öffnet eine Datei.

Argumente:

FILENAME:

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

FORCE/S:

Der Requester 'You made x changes' wird unterdrückt.

Rückgabewerte:

rc = 5, falls das Laden scheiterte.

Siehe auch:

CLEAR, SAVE, SAVEAS

1.41 FileXARexx.guide/PASTE

PASTE

=====

Aufbau:

PASTE

Funktion:

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

Siehe auch:

Menu: 'Edit/Paste', SETBLOCKMARK

1.42 FileXARexx.guide/POSITION

POSITION

=====

Aufbau:

POSITION SOF/S,EOF/S,SOL/S,EOL/S

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.

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

1.43 FileXARexx.guide/PRINT

PRINT
=====

Aufbau:
 PRINT

Funktion:
 Druckt die Datei aus.

1.44 FileXARexx.guide/PRINTBLOCK

PRINTBLOCK
=====

Aufbau:
 PRINTBLOCK

Funktion:
 Druck den Block aus dem Blockspeicher.

1.45 FileXARexx.guide/QUIT

QUIT
=====

Aufbau:
 QUIT FORCE/S

Funktion:
 Beendet das Programm.

Argumente:
 FORCE/S:
 Der Requester 'You made x changes' wird unterdrückt.

1.46 FileXARexx.guide/REDO

REDO
=====

Aufbau:
 REDO NUMBER/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.

1.47 FileXARexx.guide/REQUESTFILE

REQUESTFILE
=====

Aufbau:

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

Funktion:

Öffnet eine 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'

1.48 FileXARexx.guide/REQUESTNOTIFY

REQUESTNOTIFY
=====

Aufbau:

```
REQUESTNOTIFY PROMPT
```

Funktion:

Öffnet eine einfachen Mitteilungsrequester mit einem Gadget.

Argumente:

PROMPT:

Titeltext.

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

Beispiel:

```
/* Requester mit 4 Gadgets öffnen */
REQUESTRESPONSE BUTTONS '"Gadget 1|Gadget2|Gadget 3|Gadget0"'

say 'Du hast Gadget Nummer' rc 'ausgewählt.'
```

1.50 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.

1.51 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.

1.52 FileXARexx.guide/SAVE

SAVE

====

Aufbau:

SAVE

Funktion:

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

Rückgabewerte:

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

1.53 FileXARexx.guide/SAVEAS

SAVEAS

=====

Aufbau:

SAVEAS NAME

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.

Rückgabewerte:

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

1.54 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.

1.55 FileXARexx.guide/SAVEBLOCKCLIP

SAVEBLOCKCLIP
=====

Aufbau:
 SAVEBLOCKCLIP

Funktion:
 Kopiert den Block aus dem Blockspeicher ins Clipboard.

Siehe auch:
 LOADBLOCKCLIP

1.56 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.

1.57 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.

1.58 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.

1.59 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.

1.60 FileXARexx.guide/SETATTR

SETATTR
=====

Aufbau:

SETATTR OBJECT/A, FIELD, 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.

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.'
meinestemvariable.HEXFILLSTRING = '08 15'

SETATTR PROJECT FROMSTEM meinestemvariable
```

Siehe auch:
Attribute, GETATTR

1.61 FileXARexx.guide/SETBLOCK

SETBLOCK
=====

Aufbau:
SETBLOCK BLOCK/A

Funktion:
Setzt den aktuellen Block im Blockspeicher.

Argumente:
BLOCK/A:
Hexzahlenfolge.

Beispiel:
SETBLOCK '"0071 12feee de"'

1.62 FileXARexx.guide/SETBLOCKMARK

SETBLOCKMARK
=====

Aufbau:
SETBLOCKMARK

Funktion:

Wechselt zwischen dem Blockmarkieren-Modus und dem normalen Modus.

Siehe auch:

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

1.63 FileXARexx.guide/SETBOOKMARK

SETBOOKMARK

=====

Aufbau:

SETBOOKMARK NUMBER/N

Funktion:

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

Argumente:

NUMBER/N:

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

Siehe auch:

GOTOBOOKMARK

1.64 FileXARexx.guide/SETBYTES

SETBYTES

=====

Aufbau:

SETBYTES POS/N/A, HEXSTRING/A

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.

Beispiel:

SETBYTES 200 '"4b6c 6161 73"'

Siehe auch:

GETBYTES

1.65 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.

1.66 FileXARexx.guide/SIZEWINDOW

SIZEWINDOW
=====

Aufbau:

SIZEWINDOW HEIGHT/N,WIDTH/N

Funktion:

Die Höhe und die Breite des FileXfensters werden geändert.

Argumente:

HEIGHT/N:

Neue Höhe.

WIDTH/N:

Neue Breite des Fensters.

1.67 FileXARexx.guide/UNDO

UNDO
=====

Aufbau:

UNDO NUMBER/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.

1.68 FileXARexx.guide/WINDOWTOBACK

WINDOWTOBACK
=====

Aufbau:
WINDOWTOBACK

Funktion:
Das Fenster wird in den Hintergrund gebracht.

1.69 FileXARexx.guide/WINDOWTOFRONT

WINDOWTOFRONT
=====

Aufbau:
WINDOWTOFRONT

Funktion:
Das Fenster wird in den Vordergrund gebracht.

1.70 FileXARexx.guide/ZIPWINDOW

ZIPWINDOW
=====

Aufbau:
ZIPWINDOW

Funktion:
Gleiche Funktion wie das Zipgadget des FileXfensters.

1.71 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.

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.

"REGNAME", Text, nur lesbar:

Name des Benutzers, auf den diese Version registriert ist.

PROJECT-Objekt

=====

"AREXX", Text, nur lesbar:

Name des ARexxports.

"FILENAME", Text:

Aktueller Dateiname mit gesamtem Pfad.

"PATH", Text:

Pfad der aktuellen Datei.

"FILE", Text:

Dateiname ohne Pfad.

"CHANGES", Zahl, nur lesbar:

An Zahl der Änderungen am File.

"PRIORITY", Zahl:

Taskpriorität von FileX.

"CURSORPOS", Zahl, nur lesbar:

Aktuelle Cursorposition.

"MARKPOS", Zahl, nur lesbar:

Position des Markierungsanfangs.

"FILELEN", Zahl, nur lesbar:

Länge der aktuellen Datei.

"EDITABLE", Bool:
TRUE, falls das File edierbar ist.

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

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

"SEARCHSTRING", Text:
Suchen-Zeichenkette.

"REPLACESTRING", Text:
Ersetzen-Zeichenkette.

"HEXSEARCHSTRING", Text:
Suchen-Hexzahlenfolge.

"HEXREPLACESTRING", Text:
Ersetzen-Zeichenkette.

"STRINGSEARCH", Bool:
TRUE, falls zuletzt nach einer Zeichenkette gesucht wurde.

"CASESENSITIVESEARCH", Bool:
TRUE, falls zuletzt mit Großkleinunterscheidung gesucht wurde.

"WILDSEARCH", Bool:
TRUE, falls zuletzt mit Muster gesucht wurde.

"FILLSTRING", Text:
Füll-Zeichenkette.

"HEXFILLSTRING", Text:
Füll-Hexzahlenfolge.

"STRINGFILL", Bool:
TRUE, falls zuletzt mit einer Zeichenkette gefüllt wurde.

"DISPLAYTYP", Zahl:
Aktuell eingestellter Displaytyp.

"MAXUNDOMEMSIZE", Zahl:
Größe des maximalen Undospeichers.

"MAXUNDOLEVEL", Zahl:
Größe der maximalen Undoanzahl.

"ALTJUMP", Zahl:
Sprungweite in Zeilen von Alt-CURSORUP/DOWN.

"SCROLLBORDER", Zahl:
Abstand in Zeilen zur oberen/unteren Begrenzung, ab der das Fenster gescrollt wird.

"CLIPBOARDUNIT", Zahl:
Nummer des Clipboards für den Austausch von Daten aus dem

Blockspeicher ins Clipboard.

"COMMANDSHELLWINDOW", Text:
Beschreibung des Commandshellwindows.

"COMMANDWINDOW", Text:
Beschreibung des Ein-/Ausgabefenster für ARexxscripts.

"OVERWRITE", Bool:
TRUE, falls keine Abfrage vor dem Speichern einer existierenden
Datei kommen soll.

"USEASL", Bool:
TRUE, falls 'asl.library' und 'intuition.library' für die
Requester gebraucht werden sollen. Andernfalls wird die
'regtools.library' gebraucht.

"DISPLAY", Zahl:
Bestimmt die Aufteilung der Anzeige: 1 = nur Hexdarstellung, 2 =
nur Ascii-darstellung, 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.

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.

"SCREEN", Text, nur lesbar:

 Name des öffentlichen Bildschirms, auf dem das Fenster geöffnet
 ist.