

ARexx-Befehle

COLLABORATORS

	<i>TITLE :</i> ARexx-Befehle		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 26, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ARexx-Befehle	1
1.1	MicroPaint - ARexx-Befehle Guide 1.0.4	1
1.2	Introduction	1
1.3	Ein Beispiel-Skript	1
1.4	ARexx Commands	2
1.5	Mitgelieferte ARexx-Skripte	6
1.6	Index	7

Chapter 1

ARexx-Befehle

1.1 MicroPaint - ARexx-Befehle Guide 1.0.4

Einführung

Ein Beispiel-Skript

ARexx-Befehle

Mitgelieferte ARexx-Skripte

1.2 Introduction

Was ist ARexx?

ARexx ist eine Sprache, mit der man Anwendungsprogramme kontrollieren kann. Zu diesem Zweck werden Befehle an den ARexx-Port der Anwendung geschickt, den diese zuvor einrichten muss.

Die ARexx-Fähigkeiten von MicroPaint

Der zur Verfügung gestellte ARexx-Port heißt 'MicroPaint'. Die Standard-Extension für ARexx-Skripte ist ".mprx".

MicroPaint kommt bereits mit einigen
Beispiel-Skripten

.

Einige Befehle liefern über die ARexx-Variable result ein Ergebnis, falls dies angefordert wurde. Befehle ohne Ergebnis liefern lediglich einen Leerstring; andernfalls wird die gewünschte Information oder "ERROR" bei einem Fehler zurückgegeben.

1.3 Ein Beispiel-Skript

Das folgende Skript macht das aktuelle Bild 10 % heller:

```

                                                    Zeilennummer (nicht eingeben)
/* Calibration */                                1
address 'MicroPaint'                             2
'CALIBRATE' 'RGB' '1.1' 0 255 0                 3
```

Zeile eins ist nur ein Kommentar - dies wird vom ARexx-Interpreter "RexxMast" so erwartet.

Zeile zwei sagt ARexx, dass die nächsten Befehle an den Port 'MicroPaint' gesendet werden sollen.

Zeile drei führt die Kalibrierung aus. Hierfür sind mehrere Argumente erforderlich, die ARexx an MicroPaint übergibt: Maske, Faktor, Bereich der betroffenen Farbregister sowie ein optionaler Wert für Register, die den Wert 0 enthalten.

Hinweise:

- Alle Argumente müssen durch Leerzeichen getrennt werden.
- Zeichenketten werden am besten in einfache Anführungszeichen gesetzt. Dies ist auf jeden Fall erforderlich, falls diese Leerzeichen enthalten.

1.4 ARexx Commands

Hinweise:

- Alle Parameter müssen in der angegebenen Reihenfolge erscheinen.
- Optionale Parameter werden in der Beschreibung in eckige Klammern ([..]) gesetzt. Diese dürfen jedoch nicht mit eingegeben werden.
- Wenn Argumente Leerzeichen enthalten, müssen sie in einfache und doppelte Anführungszeichen eingefasst werden.

Beispiel: 'TEXT' x y '"Dies ist ein Beispiel."'

- Ergebnisse werden an die ARexx-Variable result übergeben, falls das Skript vor der Zeile "address 'MicroPaint'" die Zeile "options results" enthält.

Bei einem Fehler wird die Zeichenkette "ERROR" geliefert.

ABOUT

Dieser Befehl hat keine Argumente.

ABOUT zeigt den Info-Requester an.

CALIBRATE

Dieser Befehl hat folgende Argumente:

Maske	Dies ist eine beliebige Kombination der Buchstaben R, G und B, die für die zu verändernden Farbkomponenten rot, grün und blau stehen.
Faktor	Mit diesem Faktor werden die aktuellen Inhalte der Farbregister multipliziert (bzw. durch dessen Absolutwert dividiert, falls er negativ ist).
Register1 Register2	Diese Werte geben den Bereich der zu verändernden Farbregister an.
[Nullwert]	Diesen Wert erhalten Farbregister, die den Wert 0 enthalten. Falls der Wert nicht angegeben ist, wird null eingesetzt.

CALIBRATE entspricht dem Menübefehl "Tools/Kalibrieren".

CLS

Dieser Befehl hat folgendes Argument:

[Farbe]	Mit dieser Farbe (im Bereich $0..2^{\text{Bitplanes}}-1$) wird der Bildschirm gelöscht. Wird die Farbe nicht angegeben, wird die Hintergrundfarbe benutzt.
---------	-------------------------------------------------------------------------------------------------------------------------------------------------------------

CLS löscht den Bildschirm.

COLOUR

Dieser Befehl hat folgendes Argument:

Farbe	Die zu verwendende Farbe.
-------	---------------------------

CLS legt das zu benutzende Farbregister fest.

CRSRSIZE

Dieser Befehl hat folgende Argumente:

Pinselgröße	Die Größe des Standardpinsels wird auf diesen Wert gesetzt (falls er größer 0 ist).
[Linienstärke]	Die Linienstärke wird auf diesen Wert gesetzt (falls er größer 0 ist).
	Falls keine Linienstärke angegeben ist, wird der Wert der Pinselgröße verwendet.

Beispiel: CRSRSIZE 15 setzt beide Werte auf 15.

FONT

Dieser Befehl hat folgende Argumente:

Name	Der zu verwendende Zeichensatz, z. B. 'topaz.font'.
Größe	Die zu verwendende Zeichensatzgröße.
[Stil]	Der Stil, z. B. 0 für normale Schrift.
[Jam]	Der Zeichenmodus (drawmode).
[Pen1]	Die Registernummer der Vordergrundfarbe.
[Pen2]	Die Registernummer der Hintergrundfarbe.

Hinweis: Die Parameter müssen in der angegebenen Reihenfolge benutzt werden, z. B. kann man Jam nur angeben, wenn auch Stil schon definiert wurde. Wird ein optionaler Parameter nicht angegeben, werden die aktuellen Einstellungen benutzt.

FONT legt den Zeichensatz fest.

GETxxx

Diese Befehle haben keine Argumente und geben alle ein Ergebnis zurück.

Funktion	Ergebnis
GETCOLOUR	aktuelle Farbe
GETX	Cursorposition
GETY	Cursorposition
GETMAXX	Bildschirmbreite
GETMAXY	Bildschirmhöhe
GETDEPTH	Bildschirmtiefe
GETID	Bildschirmmodus
GETMODE	Zeichenmodus
GETFILENAME	aktueller Dateiname oder ""

GETxxx gibt eine Information zurück.

IDNAME

Dieser Befehl hat folgendes Argument:

ID	Dieser Wert definiert die View Mode ID, deren Namen Sie ermitteln möchten.
----	----------------------------------------------------------------------------

Beispiel: IDNAME 0 -> LoRes

IDNAME ermittelt den Namen zu einer Modus ID.

INPUT

Dieser Befehl hat folgende Argumente:

[Titel]	Dieser Wert definiert den Fenstertitel und darf maximal 20 Zeichen lang sein.
[Frage 1]	Diese beiden Argumente definieren die Fragen, die in dem Fenster gestellt werden sollen und dürfen jeweils nicht länger als 25 Zeichen sein.
[Frage 2]	
[Vorgabewert]	Dieser Wert wird in das Eingabefeld geschrieben.

INPUT übergibt den eingegebenen String.

INTEGERFORMAT

Dieser Befehl hat folgendes Argument:

Nummer	Dieser Wert ist die ganze Zahl, die Sie formatieren möchten. Zur Formatierung wird - falls verfügbar - die "locale.library" benutzt.
--------	--------------------------------------------------------------------------------------------------------------------------------------

Beispiel: INTEGERFORMAT 1234 -> 1.234

INTEGERFORMAT formatiert eine natürliche Zahl.

MODENAME

Dieser Befehl hat folgendes Argument:

Modus	Dieser Wert definiert die Nummer des Modus, dessen Namen Sie ermitteln möchten.
-------	---------------------------------------------------------------------------------

MODENAME ermittelt den Namen eines Zeichenmodus.

OPEN

Dieser Befehl hat folgendes Argument:

Name	Dies ist der Name der zu öffnenden Datei. (Geben Sie "?" an, wenn ein Dateirequester geöffnet werden soll.)
------	-------------------------------------------------------------------------------------------------------------

OPEN öffnet eine Datei und liefert als Ergebnis deren Namen.

PLOT

Dieser Befehl hat folgende Argumente:

X	Dieser Wert definiert die horizontale Position.
Y	Dieser Wert definiert die vertikale Position.
[Farbe]	Dieser optionale Parameter bestimmt die zu verwendende Farbe. Wird der Wert nicht angegeben, benutzt MicroPaint die aktuelle Farbe.

PLOT zeichnet genau einen Punkt.

PRINT

Dieser Befehl hat keine Argumente.

PRINT druckt das aktuelle Bild aus.

QUIT

Dieser Befehl hat keine Argumente.

QUIT beendet MicroPaint ohne weitere Abfrage.

REQUEST

Dieser Befehl hat folgende Argumente:

[Titel]	Dieser Wert definiert den Titel des Requesters. Die Übergabe eines leeren Strings ("") führt zu der Verwendung eines Standardtitels.
[Text]	Diese Zeichenkette definiert den darzustellenden Text. Fügen Sie einen senkrechten Strich () ein, um eine neue Zeile zu beginnen.
[Knöpfe]	Dieser Wert definiert die darzustellenden Knöpfe. Mehrere Knöpfe sind durch senkrechte Striche zu trennen.

Beispiel: 'REQUEST' "Demo" ' "Wählen Sie|eine Zahl:" "1|2|3|keine"

REQUEST stellt einen Requester dar und übergibt als Ergebnis die Nummer des gewählten Knopfes bzw. 0, falls der rechte angeklickt wurde.

TEXT

Dieser Befehl hat folgende Argumente:

X	Dieser Wert definiert die horizontale Position.
Y	Dieser Wert definiert die vertikale Position.
Text	Dies ist der zu druckende Text.
[Abstand]	Dieser optionale Parameter definiert den Zeichenabstand. Wird er nicht angegeben, gelten die aktuellen Einstellungen.

TEXT gibt einen Text auf dem Bildschirm aus.

1.5 Mitgelieferte ARexx-Skripte

Brighter.mprx

Dieses Skript macht das Bild 10 % heller.

ClearScreen.mprx

Dieses Skript löscht den Bildschirm.

Hinweis: Dies ist nur eine einfache Demonstration des CLS-Befehls.

Darker.mprx

Dieses Skript macht das Bild 10 % heller.

LessXXX.mprx

Dieses Skript erhöht den Anteil der betroffenen Farbkomponente um 10 %.

MoreXXX.mprx

Dieses Skript verringert den Anteil der betroffenen Farbkomponente um 10 %.

Quit.mprx

Dieses Skript beendet MicroPaint.

Hinweis: Dies ist nur eine einfache Demonstration des QUIT-Befehls.

Test.mprx

Dieses Skript testet alle MicroPaint-Befehle. Es sollte im CLI gestartet werden, da es zum Teil Ergebnisse ausgibt.

TextDemo.mprx

Dieses Skript schreibt eine Text in Times-Schrift auf den Bildschirm.

Hinweis: Dies ist nur eine einfache Demonstration der Befehl FONT und TEXT.

1.6 Index

A

ARexx-Befehle

B

Beispiel-Skripten

E

Ein Beispiel-Skript

Einführung

M

Mitgelieferte ARexx-Skripte