

QuickMsg

Axel Greve

COLLABORATORS

	<i>TITLE :</i> QuickMsg		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Axel Greve	August 13, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 QuickMsg	1
1.1 QuickMsg.rexx	1
1.2 Systemvoraussetzungen	2
1.3 Installation	2
1.4 Benutzung	2
1.5 Beispiele	3
1.6 Beispiele 1 & 2	4
1.7 Beispiel 3	4
1.8 Beispiel 4	4
1.9 Beispiel 5	5
1.10 Beispiele 6 & 7	5
1.11 Beispiel 8	5
1.12 Beispiel 9	6
1.13 Beispiel 10	6
1.14 Hinweise	7
1.15 Rechtliches	7
1.16 Danksagungen	7
1.17 Geschichtliches	8
1.18 Autor	8

Chapter 1

QuickMsg

1.1 QuickMsg.rexx

QuickMsg.rexx 1.0

Programm und Dokumentation von A. Greve

QuickMsg.rexx wurde geschrieben, um eine Nachricht für eine wählbare Zeit in einem eigenen Fenster auf einem (ebenfalls wählbaren) PubScreen anzuzeigen, ohne eine Bestätigung eingeben zu müssen. Das Fenster besitzt daher keine Buttons; es blockiert aber nicht und verschwindet nach einiger Zeit auf jeden Fall. Somit können Nachrichten wie z. B.

»Konnte Voreinstellungen nicht finden, verwende Standardwerte« ausgegeben werden, ohne dass der Ablauf unterbrochen wird und Sie eine Meldung bestätigen müssen.

Systemvoraussetzungen

Installation

Benutzung

Beispiele

Beispiele 1 & 2

-> D=..., P=...

Beispiel 3

-> FROM Dateiname

Beispiel 4

-> \n (Zeilenumbruch)

Beispiel 5

-> " (Anführungszeichen)

Beispiele 6 & 7

-> Batch-Anwendung, FILE Dateiname

Beispiel 8

-> M=...

Beispiel 9
-> *FROM, *FILE

Beispiel 10
-> W=..., H=...

Hinweise

Rechtliches

Danksagungen

Geschichtliches

Autor

Für die Darstellung der Textausgabe und als Ersatz für Anführungszeichen ↔

werden in dieser Beschreibung die Symbole » und « verwendet. Bei der Ausgabe des Textes erscheinen diese Zeichen selbstverständlich nicht in der Anzeige, sie sollen nur den linken und rechten Rand der Anzeige symbolisieren. Als Ersatz für Anführungszeichen dienen sie, um eine Abgrenzung zum tatsächlichen " zu erreichen.

1.2 Systemvoraussetzungen

- * rexxsupport.library v34.9+, Bestandteil der ARexx-Installation
 - * triton.library v1.3+, von Stefan Zeiger, Aminet: dev/gui
 - * tritonrexx.library v37.6+, von Jürgen Kohrmeyer, Aminet: util/rexx
- Wenn die angegebenen Libraries nicht installiert sind, wird der Text in einem CLI-Fenster ausgegeben.
- * Kickstart/Workbench 2.04+ (siehe auch die Voraussetzungen der verwendeten Libraries in deren Dokumentationen).
 - * REXXmast muss gestartet sein.

1.3 Installation

Kopieren Sie das Programm dorthin, wo Sie Ihre ARexx-Skripte abspeichern (normalerweise REXX:).

Kopieren Sie diese Dokumentation dorthin, wo Sie gewöhnlich Ihre Dokumentationen aufbewahren.

1.4 Benutzung

Das Programm kann nur von CLI/Shell benutzt werden.

Starten Sie das Programm mit "rx [Pfadname]QuickMsg[.rexx] [Argument]". Pfadname (endet mit ":" oder "/") und die Erweiterung ".rexx" werden im Regelfall (wenn das Skript nach REXX: kopiert wurde) nicht benötigt. Mit ? als Argument zeigt das Programm eine Kurzhilfe an.

Unter

Beispiele

finden Sie einige Anwendungen für QuickMsg, die die Argumentübergabe im Detail beschreiben.

Das Programm kann auf 3 verschiedene Arten aufgerufen werden:

- 1.) rx QuickMsg.rexx <Text> [D=<Sekunden>] [P=<PubScreen>] [M=<Textformat>] [W=< ←
Breite>] [H=<Höhe>]
- 2.) rx QuickMsg.rexx FILE <Dateiname> [D=<Sekunden>] [P=<PubScreen>] [M=< ←
Textformat>] [W=<Breite>] [H=<Höhe>]
- 3.) rx QuickMsg.rexx FROM <Dateiname>

<Text> oder FILE <Dateiname> müssen immer als erstes Argument angegeben werden, die weiteren Argumente sind optional und ihre Reihenfolge ist beliebig.

Bedeutung der Argumente:

<Text>	Anzuzeigender Text
FILE <Dateiname>	Anzeige einer Textdatei, Argumente aus Kommandozeile
FROM <Dateiname>	Anzeige einer Textdatei, Argumente aus Textdatei
D=<Ganzzahl>	Anzeigedauer in Sekunden
P=<String>	Name des PubScreens, auf dem das Fenster geöffnet wird
M=<Ganzzahl>	Darstellungsmodus für den Text, siehe Beispiel 8
W=<Ganzzahl>	Textbreite in Zeichen
H=<Ganzzahl>	Texthöhe in Zeilen

1.5 Beispiele

Die Anwendung von [Argument] lässt sich am besten durch ←
Beispiele erläutern:

Beispiele 1 & 2

-> D=..., P=...

Beispiel 3

-> FROM Dateiname

Beispiel 4

-> \n (Zeilenumbruch)

Beispiel 5

-> " (Anführungszeichen)

Beispiele 6 & 7

-> Batch-Anwendung, FILE Dateiname

Beispiel 8

-> M=...

Beispiel 9

-> *FROM, *FILE

Beispiel 10

-> W=..., H=...

1.6 Beispiele 1 & 2

```
rx QuickMsg Display this for 10 secs D=10 P=DOPUS.1
```

zeigt für 10 Sekunden »Display this for 10 secs« auf dem PubScreen von DirOpus an. Mit D=<Sekunden> wird die Anzeigedauer bestimmt, beim Fehlen des Arguments werden 2 Sekunden eingesetzt. P=<PubScreenname> gibt den PubScreen an. Fehlt die Angabe, wird das Fenster auf der Workbench geöffnet. PubScreen-Namen sind Case-sensitiv, es muss also auf die Groß-/Kleinschreibung geachtet werden.

```
rx QuickMsg D=4 P=y d=2 p="Test Screen"
```

zeigt für 2 Sekunden »D=4 P=y« auf der Workbench an, wenn kein PubScreen mit dem Namen »Test Screen« existiert. Es wird das letzte Auftreten von D=... und P=... ausgewertet, nachfolgender Text wird ignoriert. Groß-/Kleinschreibung wird nicht beachtet. Wenn der PubScreen-Name Leerzeichen enthält, muss er in Anführungszeichen gefasst werden. Die Zuweisungen für D und P müssen immer ohne Leerzeichen erfolgen, D= 2 oder D = 2 oder D =2 werden nicht ausgewertet.

1.7 Beispiel 3

```
rx QuickMsg FROM Ram:T/test
rx QuickMsg fRoM "Ram Disk:T/test"
```

holt den anzuzeigenden Text und die Einstellungen für Zeitdauer und PubScreen nicht aus den Argumenten sondern aus der übergebenen Datei. Beide angegebenen Beispiele verweisen auf die selbe Datei, wenn das Laufwerk RAM: den Namen »Ram Disk:« hat. Bei Leerzeichen im Dateinamen muss er in Anführungszeichen gefasst werden. Zwischen FROM (nicht case-sensitiv) und dem Dateinamen muss ein Leerzeichen stehen.

Im Unterschied zu FILE (siehe
Beispiel 6

) wird nur die erste Zeile der Datei eingelesen und der Rest der Befehlszeile ignoriert.

1.8 Beispiel 4

```
rx QuickMsg a\nb\n
```

Soll ein mehrzeiliger Text angezeigt werden, muss an der Stelle des Zeilenumbruchs ein »\n« eingefügt werden. Obiges Beispiel zeigt in der ersten Zeile ein »a«, in der zweiten ein »b« und eine leere dritte Zeile an.

1.9 Beispiel 5

```
rx QuickMsg "a"b'c'
```

Da TritonRexx zur Definition von Strings Anführungszeichen »"« benutzt, dürfen diese im anzuzeigenden Text nicht auftreten. QuickMsg wandelt sie in »'« um, damit das Programm nicht mit Fehler 18 (Ungültiges Argument für Funktion) abbricht. Obiger Aufruf würde also »'a'b'c'« anzeigen.

1.10 Beispiele 6 & 7

```

                .key Arg1/A,Arg2
.brak {
.ket }
.def Arg2 Workbench
Info {Arg1} ALL >T:test
rx QuickMsg FILE T:test D=5 P={Arg2}

```

Dieses Beispiel zeigt die Anwendung in einem Batch mit Argumentübergabe. Wenn Arg1 gleich RAM: und Arg2 gleich DOPUS.1 ist, wird anschließend die nach T:test umgelenkte Ausgabe des Info-Befehls (mit Informationen über RAM:) für 5 Sekunden auf dem DirOpus-PubScreen angezeigt. Für FILE gelten die gleichen Konditionen wie für FROM, siehe dazu auch

Beispiel 3

Im Gegensatz zu FROM wird mit FILE immer die ganze Datei eingelesen und die Übergabe von D=..., M=... und P=... in der Befehlszeile ausgewertet.

```

.key Arg1/A,Arg2,Arg3
.brak {
.ket }
.def Arg2 Workbench
.def Arg3 5
Echo "Aktuelle Einstellung:*n" >t:ShowAsgnTmp{$$}
Assign {Arg1} Exists >>t:ShowAsgnTmp{$$}
rx QuickMsg FILE t:ShowAsgnTmp{$$} P={Arg2} D={Arg3}
Delete t:ShowAsgnTmp{$$} Quiet

```

Noch ein ähnliches Beispiel: Dem Skript wird ein zugewiesener Pfad übergeben (z. B. S:) und QuickMsg zeigt für die Dauer von {Arg3} an, auf welchen physikalischen Pfad diese Zuweisung tatsächlich zeigt. Der Platzhalter \$\$ stellt die aktuelle Prozessnummer des CLI dar, die Datei könnte also z. B. »t:ShowAsgnTmp21« heißen. Mit diesem Anhängsel können Ausgaben aus mehreren parallel laufenden CLI's unterschieden werden.

1.11 Beispiel 8

```
rx QuickMsg Noch ein Beispiel M=2
```

zeigt »Noch ein Beispiel« in Fettschrift an.

Mit M=... kann die Darstellung des Textes beeinflusst werden:

M=0 normale Darstellung
 M=1 Text hervorgehoben
 M=2 Fettschrift
 M=3 Text mit 3D-Effekt
 M=4 Darstellung des Textes wird aus den Einstellungen von TritonPrefs für den Titel übernommen
 Andere Zahlenwerte ergeben die normale Darstellung. Bei Textzuweisungen wie M=abc wird »M=abc« angezeigt.

1.12 Beispiel 9

Sollen die Schlüsselwörter FROM und FILE als Text angezeigt werden, muss ihnen jeweils ein * vorangestellt werden:

```
rx QuickMsg *From now on...
rx QuickMsg *File deleted !!!
```

zeigt für jeweils 2 Sekunden »From now on...« und »File deleted !!!« auf der Workbench an.

1.13 Beispiel 10

Falls die Textausgabe in Breite und/oder Höhe (Angabe jeweils in ↔ Zeichen) begrenzt werden soll, können die Argumente W=<Zahl> und/oder H=<Zahl> mit angegeben werden:

```
rx QuickMsg Eingeschränkte Textausgabe W=8
rx QuickMsg Abgeschnittene Textausgabe W=6 H=3
```

zeigt nacheinander für jeweils 2 Sekunden folgende Texte auf der Workbench an:

```
»Eingesch
»ränkte T«
»extausga«
» be «

»Abgesc«
»hnitte«
»ne Tex«
```

Falls die Ausgabe des Textes nicht vorhersehbare Ergebnisse liefern könnte, wie beispielsweise mit den Argumenten FROM (

Beispiel 3
) oder FILE

(

Beispiele 6 & 7

), kann so die erzwungene Anzeige im CLI verhindert werden, wenn TritonRexx feststellt, dass das Fenster zu groß werden würde. Siehe dazu auch den zweiten Absatz unter

Hinweise

Die erste Ausgabe begrenzt nur die Breite; wenn auch die Texthöhe zu groß wird, erfolgt die Anzeige trotzdem im CLI. Bei der zweiten Ausgabe wird zusätzlich die Höhe begrenzt und der folgende Text geht verloren, wenn die Höhenbegrenzung einsetzt!

Bei Angabe beider Argumente wird zuerst die Breite und danach die Höhe berechnet.

1.14 Hinweise

Wenn Triton und TritonRexx nicht oder nicht richtig auf Ihrem System installiert sind, wird der Text im normalen CLI dargestellt. Die Beeinflussung der Textanzeige aus

Beispiel 8

ist dann nicht möglich. Da in diesem Fall die Anzeige z. B. durch die Zeilenumbrüche unübersichtlich wird, sollten Sie dieses Programm besser nicht verwenden ;-)

Wenn TritonRexx zur Laufzeit feststellt, dass das Fenster zu groß werden würde, weil der verwendete Screen nicht hoch oder nicht breit genug ist, wird der Text ebenfalls im CLI ausgegeben.

Nach dem Beenden von QuickMsg werden die programmbezogenen Fenster-Einstellungen (ENV:Triton/QuickMsg.win.1) gelöscht. Damit wird erreicht, dass sich das Fenster beim nächsten Start von QuickMsg wieder zentriert auf dem Screen öffnet und nicht die Position des ersten Starts einnimmt.

Die Darstellung eines Textfiles (FILE in

Beispiel 6

) sieht mit einem

nicht-proportionalen Font wie "topaz" besser aus. Der Grund dafür ist, dass TritonRexx die Textdarstellung zentriert und deshalb die Zeilen des Textfiles mit Leerzeichen auf die Länge der längsten Zeile justiert werden, um Linksbündigkeit zu erreichen.

QuickMsg setzt keine Beschränkungen für die Verzögerungszeit. Wenn Sie D=1000 angeben, wird der Text für 16 Minuten und 40 Sekunden angezeigt! Sie müssen selbst wissen was Sie tun...

1.15 Rechtliches

QuickMsg.rexx ist Freeware. Tun Sie damit was immer Sie wollen, aber es ist nicht gestattet, Profit damit zu machen. Auch wenn ich das Programm auf meinem System (Al200, KS 3.0, Blizzard II mit 24MB RAM) ausgiebig getestet habe, kann ich keinerlei Verantwortung für Besuche des GURUs oder Datenverluste durch Verwendung des Programms übernehmen - Sie benutzen es auf eigenes Risiko.

1.16 Danksagungen

Ich möchte Stefan Zeiger für die triton.library und Jürgen Kohrmeyer für die tritonrexx.library danken.
Ohne ihre Entwicklungen wäre dieses Programm nie möglich geworden.

1.17 Geschichtliches

v1.0 26-Sep-2002 * Initiale Version, Erstveröffentlichung

1.18 Autor

Wenn Sie dieses Programm benutzen und mögen (oder hassen), schicken Sie mir bitte eine Nachricht. Ich würde mich über Fehlermeldungen, Vorschläge zur Verbesserung oder einfach nur ein "Hallo" freuen.

eMail Axel.Greve@siemens.com
 (keine private eMail, also bitte Sorgfalt walten lassen :-)

Snail mail Axel Greve
 Hochfeld 26
 D-22607 Hamburg
 Deutschland

Senden Sie bitte Flames, Bomben etc. >NIL:

Glückliches & erfolgreiches "QuickMessaging",
Axel

--- Let's keep the Amiga spirit alive !!! ---
