

**TritonRexx**

**COLLABORATORS**

	<i>TITLE :</i> TritonRexx		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

**REVISION HISTORY**

<i>NUMBER</i>	<i>DATE</i>	<i>DESCRIPTION</i>	<i>NAME</i>

# Contents

<b>1</b>	<b>TritonRexx</b>	<b>1</b>
1.1	TritonRexx.guide	1
1.2	TritonRexx.guide/Übersicht	2
1.3	TritonRexx.guide/Copyright	3
1.4	TritonRexx.guide/History	4
1.5	TritonRexx.guide/Danksagungen	4
1.6	TritonRexx.guide/Adresse des Autors	5
1.7	TritonRexx.guide/Systemvoraussetzungen	5
1.8	TritonRexx.guide/Installation	6
1.9	TritonRexx.guide/Triton-Info	6
1.10	TritonRexx.guide/triton.library	8
1.11	TritonRexx.guide/Applikationen und Projekte	8
1.12	TritonRexx.guide/Taglisten verwenden	9
1.13	TritonRexx.guide/Oberflächen erstellen	11
1.14	TritonRexx.guide/Tips und Problemlösungen	13
1.15	TritonRexx.guide/FlushTRX	14
1.16	TritonRexx.guide/Applikationen	16
1.17	TritonRexx.guide/Projekte-Fenster	16
1.18	TritonRexx.guide/Menüs	19
1.19	TritonRexx.guide/Objekte und Attribute	21
1.20	TritonRexx.guide/Button	22
1.21	TritonRexx.guide/CheckBox	24
1.22	TritonRexx.guide/Cycle	24
1.23	TritonRexx.guide/DropBox	25
1.24	TritonRexx.guide/Listview	26
1.25	TritonRexx.guide/Palette	28
1.26	TritonRexx.guide/Scroller	29
1.27	TritonRexx.guide/Slider	29
1.28	TritonRexx.guide/String	31
1.29	TritonRexx.guide/FrameBox	31

---

---

1.30 TritonRexx.guide/Group . . . . .	32
1.31 TritonRexx.guide/Image . . . . .	35
1.32 TritonRexx.guide/Line . . . . .	36
1.33 TritonRexx.guide/Progress . . . . .	36
1.34 TritonRexx.guide/Space . . . . .	37
1.35 TritonRexx.guide/Text . . . . .	38
1.36 TritonRexx.guide/QuickHelp . . . . .	40
1.37 TritonRexx.guide/Füllmuster für Objekte . . . . .	40
1.38 TritonRexx.guide/Füllmuster für Fenster . . . . .	41
1.39 TritonRexx.guide/Funktionen . . . . .	43
1.40 TritonRexx.guide/TR_CreateApp() . . . . .	45
1.41 TritonRexx.guide/TR_DeleteApp() . . . . .	46
1.42 TritonRexx.guide/TR_OpenProject() . . . . .	46
1.43 TritonRexx.guide/TR_CloseProject() . . . . .	47
1.44 TritonRexx.guide/TR_LockProject() . . . . .	47
1.45 TritonRexx.guide/TR_UnlockProject() . . . . .	48
1.46 TritonRexx.guide/TR_LockScreen() . . . . .	49
1.47 TritonRexx.guide/TR_UnlockScreen() . . . . .	49
1.48 TritonRexx.guide/TR_ObtainWindow() . . . . .	50
1.49 TritonRexx.guide/TR_ReleaseWindow() . . . . .	50
1.50 TritonRexx.guide/ActivateProject() . . . . .	51
1.51 TritonRexx.guide/ProjectToFront() . . . . .	52
1.52 TritonRexx.guide/ProjectToBack() . . . . .	52
1.53 TritonRexx.guide/TR_AutoRequest() . . . . .	53
1.54 TritonRexx.guide/TR_EasyRequest() . . . . .	54
1.55 TritonRexx.guide/TR_GetAttribute() . . . . .	55
1.56 TritonRexx.guide/TR_SetAttribute() . . . . .	56
1.57 TritonRexx.guide/TR_SendMessage() . . . . .	57
1.58 TritonRexx.guide/TR_GetMsg() . . . . .	58
1.59 TritonRexx.guide/TR_ReplyMsg() . . . . .	60
1.60 TritonRexx.guide/TR_Wait() . . . . .	60
1.61 TritonRexx.guide/TR_HandleMsg() . . . . .	61
1.62 TritonRexx.guide/TR_FirstOccurance() . . . . .	62
1.63 TritonRexx.guide/TR_NumOccurances() . . . . .	63
1.64 TritonRexx.guide/TR_GetLastError() . . . . .	63
1.65 TritonRexx.guide/TR_GetErrorString() . . . . .	64
1.66 TritonRexx.guide/ASL_RequestFile() . . . . .	64
1.67 TritonRexx.guide/ASL_RequestFont() . . . . .	67
1.68 TritonRexx.guide/CreateImage() . . . . .	69

---

---

1.69	TritonRexx.guide/DeleteImage()	70
1.70	TritonRexx.guide/ImageWidth()	71
1.71	TritonRexx.guide/ImageHeight()	72
1.72	TritonRexx.guide/TritonRexxVersion()	72
1.73	TritonRexx.guide/TritonVersion()	73
1.74	TritonRexx.guide/Index	73

---

# Chapter 1

## TritonRexx

### 1.1 TritonRexx.guide

```
#####
```

```
'tritonrexx.library' 37.6 (17.9.95)
```

```
Copyright (C) 1994,95 Jürgen Kohrmeyer, <J_Kohrmeyer@wilam.north.de>
```

```
#####
```

Allgemeines

Übersicht

Kurze Vorstellung der Library

Copyright

Copyright und sonstige rechtliche Dinge

History

Geschichtliche Entwicklung der Library

Danksagungen

Ich möchte mich bedanken bei...

Adresse des Autors

Wohin man Kommentare und Bugreports schickt

Systemvoraussetzungen

68040, 18MB Ram ;-)

Installation

Installation der Library

Triton-Info

Informationen über 'Triton'

Grundlagen

---

triton.library  
Gemeinsamkeiten mit der 'triton.library'

Applikationen und Projekte  
Die Grundlagen für ein Programm

Taglisten verwenden  
Wie Taglisten funktionieren

Oberflächen erstellen  
Wie ein Fenster aufgebaut wird

Tips und Problemlösungen  
Was tun wenn...

FlushTRX  
Befehl zum Entfernen von Applikationen

## Programmieren

Applikationen  
Eigenschaften von Applikationen

Projekte-Fenster  
Eigenschaften von Projekten/Fenstern

Menüs  
Eigenschaften von Menüs

Objekte und Attribute  
Menüs, Gadgets, Texte und Images

Funktionen  
Funktionen der Library

## Sonstiges

Index  
Stichwortverzeichnis

## 1.2 TritonRexx.guide/Übersicht

### Übersicht

\*\*\*\*\*

Die 'tritonrex.library' ist eine ARexx-Funktionslibrary. Sie enthält Funktionen zur Programmierung von Benutzeroberflächen mit ARexx, zur Erstellung der Oberflächen wird die 'triton.library' verwendet. Siehe

---

Triton-Info

.

- \* Alle Features der 'triton.library V1.3' werden voll unterstützt
- \* Einfache Programmierung von Fenstern, Menüs, Gadgets und Requestern
- \* Datei-, und Fontrequester mit der 'asl.library'
- \* Einbindung von beliebigen Grafiken (Kickstart 3.0 erforderlich)
- \* Laden von Grafikdaten mit der 'datatypes.library'

### 1.3 TritonRexx.guide/Copyright

Copyright und sonstige rechtliche Dinge

\*\*\*\*\*

Die 'tritonrexx.library' ist FREEWARE, das Copyright bleibt weiterhin beim Autor Jürgen Kohrmeyer. Das Archiv darf frei kopiert werden, solange es unverändert weitergegeben, und nur eine geringe Kopiergebühr erhoben wird. Das unveränderte Archiv darf in Mailboxen zum Download angeboten werden, es ist jedoch nicht erlaubt Geld dafür zu verlangen oder dem Archiv Werbetexte hinzuzufügen. Das Archiv darf in Public-Domain Sammlungen aufgenommen werden, es darf jedoch nicht auf Disketten vertrieben werden die mehr als 4,- DM kosten! Der Vertrieb des Archivs auf 'CD-ROM' ist nur erlaubt, wenn der Preis für eine CD 30,- DM nicht übersteigt!

Die Weitergabe der 'tritonrexx.library' zusammen mit anderen Programmen ist nur gestattet, wenn es sich um 'PUBLIC DOMAIN' oder 'FREEWARE' handelt. Bei allen anderen Arten von Software ist die Genehmigung des Autors unbedingt erforderlich. In jedem Fall muß bei der Veröffentlichung ein Hinweis auf die 'tritonrexx.library', den Autor und die Bezugsquelle für das Komplettarchiv gegeben werden. Siehe

Adresse des Autors

.

Der Vertrieb der 'tritonrexx.library' zu kommerziellen Zwecken jeglicher Art ist nur mit Genehmigung des Autors gestattet!

Der Autor gibt keine Garantie für die Zuverlässigkeit der im Archiv enthaltenen Programme und Dateien! Sie benutzen die Dateien auf eigene Gefahr! Der Autor kann nicht für irgendwelche Schäden haftbar gemacht werden, die durch die Anwendung der Programme oder Dateien entstehen!

'triton.library'

=====

Zur Erstellung und Auswertung der Oberflächen wird die 'triton.library' verwendet, ein großer Teil dieser Anleitung besteht aus der Übersetzung der 'Triton-Anleitung'!



Die 'triton.library' ist Copyright (C) by Stefan Zeiger.

Siehe

Triton-Info

.

ARexx-Skripte und sonstige Programme

=====

Im Archiv befinden sich mehrere ARexx-Skripte und Programme:

- \* 'Break.rexx' ist Copyright (C) Nils Goers
- \* 'ChangeSTARTPRI.rexx' ist Copyright (C) Nils Goers
- \* 'Exchange.rexx' ist Copyright (C) Nils Goers
- \* 'ARexxControl' ist Copyright (C) Jürgen Kohrmeyer
- \* 'HandleCX' ist Copyright (C) Martin Berndt

## 1.4 TritonRexx.guide/History

Entstehungsgeschichte

\*\*\*\*\*

V37.1

Erste Alphaversion, Special-Herm-Edition ;-)

V37.2

Erste Betaversion

V37.6

Erste veröffentlichte Version

## 1.5 TritonRexx.guide/Danksagungen

Danksagungen

\*\*\*\*\*

Mein Dank gilt:

- \* Stefan Zeiger für seine großartige 'triton.library', ohne die dieses Projekt nicht möglich gewesen wäre.
  - \* Nils Goers für seine ARexx-Skripte, die er extra für das Archiv programmiert hat, für das Testen der Beta-Version und seine Vorschläge
-

und Ideen.

- \* Ralf Reiser, der es auch als ARexx-Einsteiger geschafft hat mit TritonRexx klarzukommen, für seine vielen Anregungen.
- \* Hermann Dörries für sein hervorragendes Mailbox-System 'Wilam'
- \* Allen, die mir ihre Ideen mitgeteilt, oder Bugs gemeldet haben.

## 1.6 TritonRexx.guide/Adresse des Autors

Adresse des Autors

\*\*\*\*\*

Der Autor ist unter folgenden Adressen erreichbar, bitte vorzugsweise die E-Mail Adresse benutzen:

Postanschrift:

Jürgen Kohrmeyer  
Oststraße 2  
49143 Bissendorf

Telefon:

+49-5402-5195

E-Mail:

j\_kohrmeyer@wilam.north.de

Support-Mailbox:

WILAM Mailboxsystem Wildeshausen

Port 1 - +49-4431-92081 : USR DualStandard V.34

Port 2 - +49-4431-92082 : ZyXel 19.2k

Port 3 - +49-4431-92082 : ISDN X.75

Login mit Username MD, Download per Menüauswahl

## 1.7 TritonRexx.guide/Systemvoraussetzungen

Systemvoraussetzungen

\*\*\*\*\*

- \* Die 'tritonrexx.library' benötigt mindestens Kickstart/Workbench 2.04.
  - \* Zur Einbindung von Grafiken wird mindestens Kickstart 3.0 benötigt.
  - \* Ein 68020+ Prozessor wird dringend empfohlen, die Library läuft aber auch auf 68000er Systemen.
  - \* Die 'triton.library', mindestens 'V1.3', wird unbedingt benötigt. Siehe
-

Triton-Info

.

\* Der ARexx-Interpreter REXXMast muß gestartet sein.

## 1.8 TritonRexx.guide/Installation

Installation

\*\*\*\*\*

Die Installation wird durch Doppelklick auf das 'Install'-Piktogramm gestartet, es wird das Standard-Installationsprogramm 'Installer' verwendet.

Außerdem muß die 'triton.library' installiert werden, dies wird nicht vom oben beschriebenen Programm durchgeführt. Im Verzeichnis 'Triton' befindet sich dazu ein Installationsprogramm.

Die Verwendung der Library setzt voraus, daß der ARexx-Interpreter zuvor gestartet wurde. Das Programm, das den Arexx-Interpreter startet, befindet sich in der Schublade SYS:System und heißt REXXMast. Falls dieses Programm in Ihrem System nicht automatisch gestartet wird müssen Sie es manuell starten. Die einfachste Methode ist es aber, das Programm REXXMast in der Batchdatei S:User-Startup aufzurufen. Dadurch wird ARexx bei jedem Start des Rechners automatisch aufgerufen und im System installiert.

Zum Starten des ARexx-Interpreters fügen Sie folgende Zeile in Ihre S:User-Startup ein:

```
SYS:System/REXXMast >NIL:
```

## 1.9 TritonRexx.guide/Triton-Info

Informationen über 'Triton'

\*\*\*\*\*

Im Verzeichnis 'Triton' des TritonRexx-Archivs ist die AmigaOS 2.04+ Version der 'triton.library' enthalten:

\*\*\*\*\*

Triton

An object oriented GUI creation system.

Release 1.3

(c) 1993-1995 Stefan Zeiger

\*\*\*\*\*

This is a minimum distribution version of Triton, the object oriented graphical user interface creation system for AmigaOS. Not included in this minimum distribution are:

- A Prefs editor for customizing Triton GUIs
- An optimized version of triton.library for OS3.0+
- Language catalogs
- Developer support material

If you want a full distribution, you may download Triton from the BBS of 'Tec Mania Systems':

```
>-----<
  Tec Mania Systems

  SysOp: Bernd Hassenzahl

  Phone: +49-6157-8742-7 (V34 28k8)
         +49-6157-8742-8 (V32t 19k2)
         +49-6157-91012-1 (ISDN X75)
         +49-6157-91012-2 (ISDN X75)
>-----<
```

Please log in as a guest and download Triton from the directory '/T-M-S/BINAER/AMIGA/WIZARD-WORKS'.

The latest Triton user archive can also be obtained via FidoNet file request as 'TRITON' from 'Publishers Treff BBS':

```
>-----<
  Publishers Treff

  SysOp: Christoph Reichert

  FidoNet: 2:244/6302.0
>-----<
```

Contacting the author:

E-Mail: s.zeiger@laren.rhein-main.de

FidoNet: Stefan Zeiger 2:244/6302.15

Mail: Stefan Zeiger  
Seligenstädter Weg 24  
D-63796 Kahl  
Germany

Voice: +49-6188-2684 (new!)

Please use EMail if possible. This is more convenient for me and your chance of getting a reply quickly is much better.

---

Live long and prosper!

## 1.10 TritonRexx.guide/triton.library

Gemeinsamkeiten mit der 'triton.library'

\*\*\*\*\*

Die 'triton.library' wird von fast allen Funktionen sehr intensiv genutzt. Viele Funktionen konvertieren lediglich die Argumente, rufen danach die entsprechende Triton-Funktion auf, und übermitteln die Rückgabewerte an das aufrufende ARexx-Skript.

Sämtliche Angaben zur Programmierung, in der Dokumentation, den Autodocs und dem C-Includefile 'libraries/triton.h', der 'triton.library' gelten auch für die 'tritonrexx.library'. Alle Funktionen, Objekte, Attribute und Flags haben die gleiche Bedeutung und sollten sich entsprechend dieser Dokumentation verhalten.

Die Schreibweise der Funktionsnamen ist mit der 'triton.library' identisch, mit dem Unterschied, daß nicht auf Groß- und Kleinschreibung geachtet wird.

Alle im C-Includefile 'libraries/triton.h' aufgeführten Makros sind in der 'tritonrexx.library' als Funktionen enthalten und können wie gewohnt verwendet werden. Auch hier wird nicht auf Groß- und Kleinschreibung geachtet.

Die Schreibweise aller Attribute und Flags ist identisch mit den Definitionen im C-Includefile 'libraries/triton.h' und den Beschreibungen in den Autodocs. Hier wird ebenfalls nicht auf Groß- und Kleinschreibung geachtet.

Siehe

Triton-Info

.

## 1.11 TritonRexx.guide/Applikationen und Projekte

Applikationen und Projekte

\*\*\*\*\*

Applikationen

Die Applikation ist die Grundlage für alle Fenster, die mit den Funktionen der 'tritonrexx.library' programmiert werden sollen. Jedes Skript muß eine solche Applikation erzeugen, bevor das erste Fenster geöffnet werden kann. Am Ende eines Skripts muß jede erzeugte Applikation wieder gelöscht werden, damit der zugehörige Speicher an das System zurückgegeben wird.

Jeder Applikation muß ein kurzer Name gegeben werden, der zur Identifikation

der Applikation dient.

## Projekte

Jedes Projekt ist ein Ableger einer bestimmten Applikation. Es beinhaltet genau ein Fenster, inclusive aller Eigenschaften, Menüs und Objekte. Jedem Projekt sollte eine einmalige ID-Nummer zugewiesen werden, diese Nummer wird von der 'triton.library' zum Speichern der Fensterposition und Größe verwendet.

Beim Öffnen eines Projekts müssen die gewünschten Eigenschaften des Fensters, alle Menüs und darzustellenden Objekte mit ihren Eigenschaften angegeben werden. Objekte sind z.B. Gadgets, Texte und Grafiken. Einige der Fenster- und Objekteigenschaften lassen sich auch nachträglich verändern.

Als erstes müssen die Eigenschaften des Fensters angegeben werden, danach die Menüs, und zum Schluß die Objekte.

## 1.12 TritonRexx.guide/Taglisten verwenden

### Taglisten verwenden

\*\*\*\*\*

Die gesamte Definition aller Applikationen und Projekte erfolgt mit Hilfe von sogenannten Taglisten. Eine Tagliste ist die Aneinanderreihung von verschiedenen Attributen, die das Aussehen und Verhalten der beschriebenen Applikation, eines Projekts oder Objekts bestimmen.

Jedes Attribut besteht immer aus einer Kennung und einer dazu gehörenden Datenangabe. Die einzelnen Attribute werden durch Leerzeichen oder Tabs getrennt hintereinander geschrieben.

Die Datenangabe hat verschiedene Formate, je nach Art der übergebenen Daten:

#### Boolscher Wert

0 (falsch) oder 1 (wahr)

#### Integerzahl

Eine Zahl zwischen -2147483648 und 2147483647

#### Zeichenkette

Eine beliebige Zeichenkette, wenn darin Leerzeichen oder Tabulatoren enthalten sind muß die Zeichenkette in Anführungszeichen gesetzt werden.

Wenn die Zeichenkette in Anführungszeichen gesetzt ist dient das Zeichen '\*' als Steuerzeichen. Der '\*' wird dann nicht in die Zeichenkette übernommen, sondern das nachfolgende Zeichen kann eine bestimmte Bedeutung haben:

\*E - ein Escape-Zeichen (ASCII 27)

\*N - ein Zeilenvorschub (ASCII 10)

\*" - ein Anführungszeichen

\*\* - ein '\*'

Alle anderen Zeichen werden ohne Änderung übernommen, um den Text "Hallo Welt" zu erhalten müßte man also "\*"Hallo Welt\*" schreiben.

#### Flags

Flags, die Eigenschaften eines Objekts angeben. Die Flags werden getrennt durch '|' hintereinandergeschrieben. Diese Angabe darf keine Leerzeichen enthalten.

#### Feld oder Liste von Zeichenketten

Der Name einer Stem-Variablen, in der die Zeichenketten enthalten sind. Die Zeichenketten müssen wie folgt gespeichert sein:

stemvar.0 - enthält die Anzahl der Zeichenketten stemvar.1 - enthält die erste Zeichenkette stemvar.2 - enthält die zweite Zeichenkette stemvar.3 - enthält die dritte Zeichenkette stemvar.n - enthält die n'te Zeichenkette

#### Zeiger

Dies betrifft Project-, Screen- und Window-Zeiger, die von anderen Funktionen zurückgegeben wurden. Die Rückgabewerte enthalten dann eine Zahl im binären Format, die so nicht als Datenangabe verwendet werden kann. Ein solcher Zeiger muß für die Angabe in einer Tagliste mit der ARexx-Funktion C2D() in eine Zahl umgewandelt werden.

#### TextAttr-Struktur

Der Name einer Stem-Variablen, die die Elemente einer TextAttr-Struktur enthält. Die Variable muß wie folgt aufgebaut sein:

stemvar.ta\_name - Enthält den Namen des Fonts, z.B. 'helvetica.font'  
 stemvar.ta\_ysize - Enthält die Größe des Fonts  
 stemvar.ta\_style - Enthält eine Zahl, die den Stil des Fonts beschreibt:

0 - normal  
 1 - unterstrichen  
 2 - fett  
 3 - unterstrichen und fett  
 4 - kursiv  
 5 - unterstrichen und kursiv  
 6 - fett und kursiv  
 7 - unterstrichen, fett und kursiv

ta\_style muß nicht angegeben werden, Standard ist 0.

#### Image

Der logische Name einer Grafik, die zuvor mit 'CREATEIMAGE()' geladen wurde.

#### TR\_Dimensions-Struktur

Eine Stem-Variable, die die Elemente einer TR\_Dimensions-Struktur enthält. Die Variable muß wie folgt aufgebaut sein:

stemvar.trd\_left - Die linke Ecke des Fensters

```

stemvar.trd_top      - Die obere Ecke des Fensters
stemvar.trd_width    - Die Breite des Fensters
stemvar.trd_height   - Die Höhe des Fensters
stemvar.trd_left2    - Die alternative linke Ecke des Fensters
stemvar.trd_top2     - Die alternative obere Ecke des Fensters
stemvar.trd_width2   - Die alternative Breite des Fensters
stemvar.trd_height2  - Die alternative Höhe des Fensters
stemvar.trd_zoomed   - Fenster gezoomt? - 0 oder 1

```

#### Ungenutzt

Die Datenangabe wird momentan nicht verwendet, muß aber angegeben werden. Am besten man gibt hier immer 0 an!

Den Abschluß einer Tagliste bildet immer 'TAG\_END' oder 'TAG\_DONE', hier muß keine Datenangabe mehr gemacht werden.

Beispiel für eine Tagliste:

```
=====
```

```

apptags = 'TRCA_Name      Wilko',
          'TRCA_LongName  Wilam-Mailboxverwaltung',
          'TRCA_Info      "Point-, Kosten- und Textverwaltung für Wilam"',
          'TRCA_Version   3.0',
          'TRCA_Release   1',
          'TRCA_Date      24.7.95',
          'TAG_END'

```

Bei diesem Beispiel dient das Komma am Ende jeder Zeile als Fortsetzungszeichen, die Anweisung wird von ARexx wie eine Zeile behandelt.

## 1.13 TritonRexx.guide/Oberflächen erstellen

Oberflächen erstellen

```
*****
```

Jedes Fenster wird in Gruppen aufgeteilt. Eine Gruppe ist ein Bereich in dem sich verschiedene Objekte befinden. Dies können auch wieder Gruppen sein, die dann andere Objekte enthalten usw.

Grundsätzlich gibt es zwei Arten von Gruppen:

```
=====
```

Horizontale Gruppen - TRGR\_Horiz

Die darin enthaltenen Objekte werden nebeneinander dargestellt. Die zweite Dimension einer solchen Gruppe ist vertikal, dies dient zur Ausrichtung der enthaltenen Objekte untereinander.

Vertikale Gruppen - TRGR\_Vert

Die darin enthaltenen Objekte werden untereinander dargestellt. Die zweite Dimension einer solchen Gruppe ist horizontal, dies dient wiederum zur Ausrichtung der enthaltenen Objekte untereinander.

Einer Gruppe können verschiedene Layout-Eigenschaften zugeordnet werden:

```
=====
```



**TRGR\_PROPSHARE**

Alle Objekte werden proportional zu ihrer minimalen Größe skaliert. Alle Leerräume bleiben dabei unverändert, das gleiche gilt auch für alle nicht skalierbaren Objekte.

**TRGR\_EQUALSHARE**

Alle Objekte haben in der primären Dimension die gleiche Größe. Alle Objekte sind mindestens so groß wie das kleinste Objekt. Für Leerräume und nicht skalierbare Objekte gilt das gleiche wie bei 'TRGR\_PROPSHARE'.

**TRGR\_PROPSPACES**

Alle Objekte, mit Ausnahme der Leerräume, behalten ihre minimale Größe. Die Leerräume werden proportional zu ihrer minimalen Größe skaliert.

**TRGR\_ARRAY**

Eine solche Gruppe bildet die Obergruppe einer Tabelle von Objekten. Um eine solche Tabelle anzulegen muß eine Obergruppe mit der Eigenschaft 'TRGR\_ARRAY' erzeugt werden, horizontal für eine spaltenorientierte, vertikal für eine zeilenorientierte Tabelle. In der 'TRGR\_ARRAY'-Gruppe können einzelne Objekte oder Untergruppen mit der entgegengesetzten Richtung enthalten sein, diese Objekte werden dann tabellarisch innerhalb der Gruppe angeordnet.

**TRGR\_INDEP**

Untergruppen einer 'TRGR\_ARRAY'-Gruppe, die das Attribut 'TRGR\_INDEP' besitzen, werden unabhängig von der Tabelle angeordnet, dies wird hauptsächlich zum Einfügen von Trennlinien verwendet.

Eigenschaften für die Ausrichtung der Objekte in der zweiten Dimension:

**TRGR\_ALIGN**

Alle in der zweiten Dimension der Gruppe skalierbaren Objekte werden auf die volle Größe der Gruppe skaliert.

**TRGR\_CENTER**

Alle in der zweiten Dimension nicht skalierbaren Objekte werden innerhalb der Gruppe zentriert.

Die Skalierbarkeit einer Gruppe kann auch abgeschaltet werden:

**TRGR\_FIXHORIZ**

Die Gruppe behält immer ihre minimale Breite.

**TRGR\_FIXVERT**

Die Gruppe behält immer ihre minimale Höhe.

Bei der Definition muß jede Gruppe mit 'TRGR\_Horiz' oder 'TRGR\_Vert' begonnen werden. Die Datenangabe enthält die der Gruppe zugeordneten Eigenschaften getrennt durch '|'. Den Abschluß einer Gruppe bildet immer 'TRGR\_End'. Die Datenangabe spielt dabei keine Rolle, muß aber vorhanden sein.

## 1.14 TritonRexx.guide/Tips und Problemlösungen

### Tips und Problemlösungen

\*\*\*\*\*

- \* Tip: Einbinden der Library mit erhöhter Priorität

Die 'tritonrexx.library' sollte mit erhöhter Priorität eingebunden werden, so daß sie von ARexx vor allen anderen Libraries nach Funktionen durchsucht wird.

In Taglisten werden sehr häufig die für die einzelnen Objekte angebotenen Makro-Funktionen verwendet, beim Interpretieren der Taglisten müssen dadurch sehr viele Funktionen aufgerufen werden. Die erhöhte Priorität kann dabei eine erhebliche Steigerung der Geschwindigkeit bewirken!

'tritonrexx.library' mit Priorität 10 einbinden:

```
-----
IF ~SHOW('LIBRARIES','tritonrexx.library') THEN DO
  IF ~ADDLIB('tritonrexx.library',10,-30,0) THEN DO
    SAY 'Kann die <tritonrexx.library> nicht öffnen!'
    EXIT(10)
  END
END
END
```

- \* Beim Aufruf von Funktionen kommt es zu Fehlern, obwohl alle Argumente richtig angegeben sind.

Dies liegt oft daran, daß andere Libraries mit gleichnamigen Funktionen durch die ARexx-Funktion 'AddLib()' oder den DOS-Befehl 'RXLib' in die Suchliste von ARexx eingebunden wurden. In diesem Fall ist es oft sehr schwierig, weiterhin beide Libraries gleichzeitig zu verwenden. Um dieses Problem zu lösen kann man erst einmal versuchen, durch unterschiedliche Prioritäten bei 'AddLib()' oder 'RXLib' die Suchreihenfolge von ARexx entsprechend zu verändern. Falls dies nicht zum gewünschten Ergebnis führt gibt es noch eine andere, eigentlich sogar bessere Methode. Alle Funktionen der 'tritonrexx.library' können auch über ein Alias aufgerufen werden, indem man dem Funktionsnamen ein 'TRX\_' voranstellt. Für die Funktion 'TR\_OpenProject()' wäre der Aliasname z.B. 'TRX\_TR\_OpenProject()'.

- \* Nach dem Abbruch eines Skripts durch Fehler bleiben Fensterleichen zurück.

Wenn ein Skript mit einem Fehler abbricht, dann bleiben die in diesem Skript bereits geöffneten Fenster funktionsunfähig auf dem Bildschirm zurück, auch Grafiken können sich noch im Speicher befinden. Zum Schließen solcher Fenster, Entfernen der zugehörigen Applikation, und Löschen von ungenutzten Grafiken, dient der mitgelieferte DOS-Befehl 'FlushTRX'.

Siehe

FlushTRX

.

Ein fertig ausgetestetes Skript sollte auf jeden Fall die verschiedenen Unterbrechungssignale von ARexx nutzen, um im Fehlerfall alle geöffneten Applikationen und Grafiken zu entfernen.

Beispiel:

```
-----
/*
** Applikation beim Auftreten eines Unterbrechungssignals
** automatisch löschen und Skript ordnungsgemäß beenden.
*/

SIGNAL ON break_c
SIGNAL ON failure
SIGNAL ON halt
SIGNAL ON ioerr
SIGNAL ON syntax

app = TR_CREATEAPP('TRCA_Name      Beispiel',
                  'TRCA_LongName  "Beispiel für SIGNAL"',
                  'TRCA_Info      "ARexx-Unterbrechungssignale nutzen"',
                  'TRCA_Version   1.0',
                  'TRCA_Release   1',
                  'TRCA_Date      30.7.95',
                  'TAG_END')

IF app ~= '00000000'x THEN DO

    /*
    ** Hier steht das Skript
    */

    CALL TR_DELETEAPP(app)
END

EXIT(0)

break_c:
failure:
halt:
ioerr:
syntax:
    SAY '+++ Fehler' rc 'in Zeile' sigl '-' ERRORTXT(rc)
    SAY SOURCELINE(sigl)

    IF app ~= '00000000'x THEN
        CALL TR_DELETEAPP(app)

    EXIT(10)
```

## 1.15 TritonRexx.guide/FlushTRX

Der DOS-Befehl FlushTRX

\*\*\*\*\*

Dieser Befehl dient zum Entfernen von Applikationen, die nach dem Abbruch eines fehlerhaften Skripts nicht wieder gelöscht wurden. Es können immer nur komplette Applikationen entfernt werden, das Schließen von einzelnen Fenstern ist nicht möglich! Ungenutzte Grafiken, die sich noch im Speicher befinden, können ebenfalls gelöscht werden. 'FlushTRX' entfernt, wenn möglich, auch die 'tritonrexx.library' aus dem Speicher.

Eine Applikation, auf die noch ein Skript zugreift, kann nicht entfernt werden. Dies ist z.B. der Fall, wenn ein Skript gerade mit 'TR\_Wait()' auf Nachrichten von der Applikation wartet. Wenn ein Skript aber mit anderen Dingen beschäftigt ist, und die Applikation deshalb momentan nicht anspricht, übernimmt 'FlushTRX' die Applikation und entfernt sie. Das Skript bricht dann beim nächsten Zugriffsversuch auf die nicht mehr bestehende Applikation mit dem Fehler 18 ab:

```
Command returned 10/18: Invalid argument to function
```

Grafiken können nur gelöscht werden, wenn sie nicht mehr in einem Fenster dargestellt werden.

Außer dem Befehl 'FlushTRX' gibt es keine systemkonforme Möglichkeit, solche Applikationen nachträglich zu entfernen! Grafiken können auch nachträglich mit 'DeleteImage()' (Siehe

```
DeleteImage()
```

```
.) gelöscht werden. Das Entfernen der
```

'tritonrexx.library' aus dem Speicher ist erst dann möglich, wenn alle Applikationen und Grafiken gelöscht sind.

#### Format

```
FlushTRX [[APPLICATION] <AppName>] [IMAGES] [ALL]
```

#### Schablone

```
FlushTRX APPLICATION, IMAGES/S, ALL/S
```

#### Argumente

APPLICATION - Der Name der zu entfernenden Applikation. Dies ist der einmalige Name, der beim Erzeugen der Applikation unter 'TRCA\_Name' angegeben wurde. Siehe

Applikationen

. Hier kann

auch ein AmigaDOS-Namensmuster angegeben werden, FlushTRX #? entfernt z.B. alle momentan nicht verwendeten Applikationen.

IMAGES - Wenn dieser Schalter angegeben ist, versucht das Programm alle noch im Speicher befindlichen Grafiken zu löschen. Die Anzahl der Grafiken, die gelöscht werden konnten, wird im Shell-Fenster ausgegeben.

ALL - Wenn dieser Schalter angegeben ist, versucht das Programm alle noch im Speicher befindlichen Applikationen und Grafiken zu löschen. Dies ist jedoch nur dann möglich, wenn kein Skript mehr auf eine Applikation zugreift.

## 1.16 TritonRexx.guide/Applikationen

Eigenschaften von Applikationen

\*\*\*\*\*

### ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TRCA_Name	Zeichenkette	Ein einmaliger Name für die Applikation, nicht länger als 20 Zeichen. Die Zeichen ' ', '/', '.' und ':' sind nicht erlaubt.
TRCA_LongName	Zeichenkette	Ein Name für die Applikation, der für Jeden sichtbar ist, bis zu 60 Zeichen.
TRCA_Info	Zeichenkette	Kurze Information über die Applikation, bis zu 2 Zeilen mit je 60 Zeichen.
TRCA_Version	Zeichenkette	Die Interne Version der Applikation.
TRCA_Release	Zeichenkette	Die Release-Version der Applikation.
TRCA_Date	Zeichenkette	Das Erstellungsdatum der Applikation, sollte im Standard-Format für Datum im Versions-String angegeben werden.

### FUNKTIONEN

Siehe

TR\_CreateApp()

.

### DEFINITIONEN

-/-

## 1.17 TritonRexx.guide/Projekte-Fenster

Eigenschaften von Fenstern

\*\*\*\*\*

### ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TRWI_Title	Zeichenkette	Fenstertitel (veränderbar)

TRWI_ScreenTitle	Zeichenkette	Bildschirmtitel bei aktivem Fenster (veränderbar)
TRWI_Flags	Flags	
	TRWF_BACKDROP	Rahmenloses Hintergrundfenster mit voller Bildschirmgröße
	TRWF_NODRAGBAR	Nicht verschiebbar
	TRWF_NODEPTHGADGET	Kann nicht nach vorne oder hinten bewegt werden
	TRWF_NOCLOSEGADGET	Ohne Schließsymbol
	TRWF_NOACTIVATE	Nicht aktiviert
	TRWF_NOESCCLOSE	Kann nicht mit ESC geschlossen werden
	TRWF_NOPSCRFBALLBACK	Nicht öffnen wenn der angegebene Bildschirm nicht existiert
	TRWF_NOZIPGADGET	Kein Zoom-Symbol
	TRWF_ZIPCENTERTOP	Gezoomtes Fenster in der Mitte der Titelzeile des Bildschirms
	TRWF_NOMINTEXTWIDTH	Fenstertitel bestimmt nicht die Breite des Fensters
	TRWF_NOSIZEGADGET	Kein Größenveränderungs-Symbol
	TRWF_NOFONTFALLBACK	Bei zu großem Font nicht topaz.font 8 verwenden
	TRWF_NODELZIP	Fenster bei DELETE nicht zoomen
	TRWF_ZIPTOCURRENTPOS	Fensterposition bleibt beim Zoomen gleich
	TRWF_APPWINDOW	Auf das Fenster kann man Icons fallen lassen
	TRWF_ACTIVATESTRGAD	Das erste Texteingabefeld im Fenster wird sofort aktiviert
	TRWF_HELP	Das Fenster sendet 'TRMS_HELP' wenn über einem Menü oder Objekt HELP gedrückt wird
	TRWF_SYSTEMACTION	Bei Diskettenwechsel sendet das Fenster eine Nachricht: 'TRMS_DISKINSERTED' 'TRMS_DISKREMOVED'
TRWI_Underscore	Zeichenkette	Zeichen, mit dem die Hotkeys markiert sind, Standard '_'
TRWI_Position	Fensterposition	
	TRWP_DEFAULT	Laß dich überraschen ;-)
	TRWP_BELOWTITLEBAR	Links unter der Titelzeile des Bildschirms
	TRWP_CENTERTOP	Zentriert, ganz oben auf dem Bildschirm
	TRWP_TOPLEFTSCREEN	In der linken oberen Ecke des Bildschirms
	TRWP_CENTERSCREEN	Zentriert auf dem Bildschirm
	TRWP_CENTERDISPLAY	Zentriert im dargestellten Bildschirmausschnitt
	TRWP_MOUSEPOINTER	Zentriert unter dem Mauszeiger

TRWI_CustomScreen	Zeiger	Eigener Bildschirm, auf dem das Fenster geöffnet werden soll
TRWI_PubScreen	Zeiger	Öffentlicher Bildschirm, auf dem das Fenster geöffnet werden soll
TRWI_PubScreenName	Zeichenkette	Namen eines öffentlichen Bildschirms, auf dem das Fenster geöffnet werden soll
TRWI_PropFontAttr	TextAttr	Stem-Variable, die eine TextAttr-Struktur für den Proportional-Font des Fensters enthält Siehe Taglisten verwenden .
TRWI_FixedWidthFontAttr	TextAttr	Stem-Variable, die eine TextAttr-Struktur für den Font mit fester Breite des Fensters enthält Siehe Taglisten verwenden .
TRWI_Backfill	Füllmuster Füllmuster für Fenster	Siehe .
TRWI_ID	Integer	ID-Nummer des Fensters, gleichartige Fenster sollten gleiche Nummern haben, sonst immer verschiedene
TRWI_Dimensions	Dimensionen	Stem-Variable, die eine TR_Dimensions-Struktur enthält. Nach dem Schließen des Fensters enthalten die Felder der hier angegebenen Variablen die letzte Position und Größe des Fensters. Siehe Taglisten verwenden .
TRWI_QuickHelp	Bool	QuickHelp ein/ausschalten

## FUNKTIONEN

WindowTitle(title)  
Fenstertitel angeben

ScreenTitle(title)  
Bildschirmtitel angeben

WindowID(id)  
ID-Nummer angeben

WindowFlags(flags)  
Flags angeben

WindowPosition(position)  
Fensterposition angeben

WindowUnderscore(underscore)  
Zeichen angeben, mit dem die Hotkeys markiert sind

WindowDimensions(TR\_Dimensions)  
Fensterdimensionen angeben

CustomScreen(screen)  
Zeiger auf einen eigenen Bildschirm

PubScreen(screen)  
Zeiger auf einen Öffentlichen Bildschirm

PubScreenName(screenname)  
Name eines öffentlichen Bildschirms

QuickHelpOn(bool)  
QuickHelp ein/ausschalten

BeginRequester(title,position)  
Markiert den Beginn eines Fensters, das mit dem Aussehen eines Requesters geöffnet werden soll

## DEFINITIONEN

EndProject  
Markiert das Ende einer Projekt-Tagliste

BeginRequesterGads  
Markiert den Beginn der Gadgets im unteren Teil eines Requesters

EndRequester  
Markiert das Ende eines Requesters

Siehe  
Füllmuster für Fenster  
.

## 1.18 TritonRexx.guide/Menüs

Menüs anlegen  
\*\*\*\*\*

## ATTRIBUTE

Kennung	Datenangabe	Beschreibung
---------	-------------	--------------

---



---

TRMN_Title	Zeichenkette	Menütitel
TRMN_Item	Zeichenkette	Ein Menüpunkt in dem zuvor angegebenen Menü. Eine Taste kann angegeben werden indem man das gewünschte Zeichen dem Text voranstellt, gefolgt von '_' und schließlich dem Text. Steuertasten können genauso angegeben werden, jedoch muß das erste Zeichen ein '_' sein.  S_Datei speichern D_Drucken  _F1_Text anzeigen _Delete_Löschen  Wenn als Text 'TRMN_BARLABEL' angegeben wird, dann ist der Menüpunkt eine Trennlinie.
TRMN_Sub	Zeichenkette	Ein Untermenüpunkt zu dem zuvor angegebenen Menüpunkt. Hier gilt das gleiche wie bei 'TRMN_Item'.
TRMN_Flags	Flags	
	TRMF_CHECKIT TRMF_CHECKED TRMF_DISABLED	Ein Menüpunkt zum abhaken Der Menüpunkt ist abgehakt Nicht anwählbar
TRAT_ID	Integer	ID-Nummer eines Menüpunkts

## FUNKTIONEN

BeginMenu(title)  
Beginn eines neuen Menüs

MenuFlags(flags)  
Menü-Flags angeben

MenuItem(text,id)  
Ein normaler Menüpunkt

MenuItemC(text,id)  
Ein Menüpunkt zum abhaken

MenuItemCC(text,id)  
Ein abgehakter Menüpunkt

MenuItemD(text,id)  
Ein normaler Menüpunkt, nicht anwählbar

BeginSub(title)

---

Beginn eines Untermenüs

SubItem(text, id)  
Ein Untermenüpunkt

SubItemD(text, id)  
Ein Untermenüpunkt, nicht anwählbar

#### DEFINITIONEN

ItemBarlabel  
Eine Trennlinie im Menü

SubBarlabel  
Eine Trennlinie im Untermenü

## 1.19 TritonRexx.guide/Objekte und Attribute

Objekte und Attribute

\*\*\*\*\*

Alle Bestandteile eines Fensters nennt man Objekte. Folgende Liste enthält alle verfügbaren Objekte mit den dazu gehörenden Eigenschaften:

Objekte, die Aktionen auslösen

Button	Allerlei Knöpfe
CheckBox	Ein/Aus Schalter
Cycle	Felder zur Mehrfachauswahl
DropBox	Flächen zum Fallenlassen von Icons
Listview	Listen zur Darstellung und Auswahl
Palette	Auswahl von Farben
Scroller	Schieberegler zum Scrollen durch Listen
Slider	Schieberegler zur Einstellung von Werten
String	Texteingabefelder

---

## Grafische Objekte

FrameBox	Einrahmen von Gruppen
Group	Gruppen
Image	Grafiken
Line	Trennlinien
Progress	Fortschrittsanzeige
Space	Leerräume zwischen Objekten
Text	Texte in verschiedenen Darstellungsarten

## Sonstiges

QuickHelp	Kurzhilfe
Füllmuster für Fenster	Fensterhintergrund füllen
Füllmuster für Objekte	Objekthintergrund füllen

**1.20 TritonRexx.guide/Button**

Button

\*\*\*\*\*

## OBJEKTNAME

TROB\_Button

## ATTRIBUTE

Kennung

Datenangabe

Beschreibung

-----  
TROB\_Button

Typ

TRBT\_TEXT

Einfacher Knopf mit Text

TRBT\_GETFILE

Datei-Request-Knopf

TRBT\_GETDRAWER

Verzeichnis-Rquest-Knopf

TRBT\_GETENTRY

Listen-Auswahl-Knopf

TRAT_Disabled	Bool	Knopf anwählbar?
TRAT_Flags	Flags	
	TRBU_RETURNOK	Anwahl auch mit RETURN
	TRBU_ESCOK	Anwahl auch mit ESC
	TRBU_SHIFTED	Tastatur nur mit SHIFT
	TRBU_UNSHIFTED	Tastatur nur ohne SHIFT
	TRBU_YRESIZE	Vertikal skalierbar
TRAT_ID	Integer	ID-Nummer des Knopfs
TRAT_Text	Zeichenkette	Text für den Knopf
TRDO_QuickHelpString	Zeichenkette	Kurzer Hilfstext

## FUNKTIONEN

Button(text, id)  
 Einfacher Knopf

ButtonR(text, id)  
 Einfacher Knopf, kann mit RETURN angewählt werden

ButtonE(text, id)  
 Einfacher Knopf, kann mit ESC angewählt werden

ButtonRE(text, id)  
 Einfacher Knopf, kann mit RETURN und ESC angewählt werden

CenteredButton(text, id)  
 Wie 'Button()', nur zentriert

CenteredButtonR(text, id)  
 Wie 'ButtonR()', nur zentriert

CenteredButtonE(text, id)  
 Wie 'ButtonE()', nur zentriert

CenteredButtonRE(text, id)  
 Wie 'ButtonRE()', nur zentriert

EmptyButton(id)  
 Ein Knopf ohne Text

GetFileButton(id)  
 Ein Datei-Request-Knopf

GetDrawerButton(id)  
 Ein Verzeichnis-Request-Knopf

GetEntryButton(id)  
 Ein Listen-Auswahl-Knopf

GetFileButtonS(s, id)  
 Wie 'GetFileButton()', nur mit Text

```
GetDrawerButtonS(s,id)
    Wie 'GetDrawerButton()', nur mit Text
```

```
GetEntryButtonS(s,id)
    Wie 'GetEntryButton()', nur mit Text
```

DEFINITIONEN

-/-

## 1.21 TritonRexx.guide/CheckBox

CheckBox

\*\*\*\*\*

OBJEKTNAME

TROB\_CheckBox

ATTRIBUTE

Kennung	Datenangabe	Beschreibung
-----		
TROB_CheckBox	Ungenutzt	
TRAT_Disabled	Bool	Schalter anwählbar?
TRAT_ID	Integer	ID-Nummer des Schalters
TRAT_Value	Bool	Eingeschaltet?
TRDO_QuickHelpString	Zeichenkette	Kurzer Hilfstext

FUNKTIONEN

```
CheckBox(id)
    Einfacher Schalter, abgeschaltet
```

```
CheckBoxC(id)
    Einfacher Schalter, eingeschaltet
```

```
CheckBoxLEFT(id)
    Wie 'CheckBox()', nur linksbündig
```

```
CheckBoxCLEFT(id)
    Wie 'CheckBoxC()', nur linksbündig
```

DEFINITIONEN

-/-

## 1.22 TritonRexx.guide/Cycle

Cycle

\*\*\*\*\*

## OBJEKTNAME

TROB\_Cycle

## ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_Cycle	Einträge	Stem-Variable, die die anwählbaren Einträge enthält
TRAT_Disabled	Bool	Auswahlfeld anwählbar? (Radio-Buttons nur ab OS3.0)
TRAT_Flags	Flags	
	TRCY_MX TRCY_RIGHTLABELS	Darstellung als Radio-Buttons Texte rechts von den Knöpfen
TRAT_ID	Integer	ID-Nummer des Auswahlfeldes
TRAT_Value	Integer	Nummer des angewählten Eintrags
TRDO_QuickHelpString	Zeichenkette	Kurzer Hilfstext

## FUNKTIONEN

CycleGadget (einträge, value, id)  
Einfaches Auswahlfeld

MXGadget (einträge, value, id)  
Auswahlfeld als Radio-Buttons

MXGadgetR (einträge, value, id)  
Wie 'MXGadget()', Texte rechts von den Knöpfen

## DEFINITIONEN

-/-

**1.23 TritonRexx.guide/DropBox**

DropBox

\*\*\*\*\*

## OBJEKTNAME

TROB\_DropBox

## ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_DropBox	Ungenutzt	
TRAT_Disabled	Bool	Dorpbox benutzbar?
TRAT_ID	Integer	ID-Nummer der Dropbox

TRDO\_QuickHelpString      Zeichenkette      Kurzer Hilfstext

#### FUNKTIONEN

DropBox(id)  
Einfache Dropbox

#### DEFINITIONEN

-/-

## 1.24 TritonRexx.guide/Listview

Listview

\*\*\*\*\*

#### OBJEKTNAME

TROB\_Listview

#### ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_Listview	Einträge	Stem-Variable, die die Einträge der Liste enthält
TRAT_Disabled	Bool	Listview aktiv?
TRAT_Flags	Flags	Typ, diese Flags schließen sich gegenseitig aus
	TRLV_READONLY	Liste ist nur lesbar
	TRLV_SELECT	Auswahl möglich
	TRLV_SHOWSELECTED	Ausgewählter Eintrag wird markiert
	Sonstige Flags	
	TRLV_NOCURSORKEYS	Cursortasten nicht verwenden
	TRLV_NONUMPADKEYS	Zehnerblock nicht verwenden
	TRLV_FWFONT	Zeichensatz mit fester Breite
	TRLV_NOGAP	Kein Leerraum unter der Liste, stattdessen am Ende der Gruppe
TRAT_ID	Integer	ID-Nummer des Listviews
TRAT_MinWidth	Integer	Minimale Breite in Zeichen (Standard sind 10 Zeichen)
TRAT_MinHeight	Integer	Minimale Höhe in Zeilen (Standard sind 4 Zeilen)
TRAT_Value	Integer	Nummer des ausgewählten Eintrags der erste Eintrag ist 0!!!

---

TRLV_Top	Integer	Nummer des obersten Eintrags in der Anzeige
TRDO_QuickHelpString	Zeichenkette	Kurzer Hilfstext

## FUNKTIONEN

ListRO(einträge, id, top)  
Einfache Liste, nur lesbar

ListSel(einträge, id, top)  
Einfache Liste, Einträge auswählbar

ListSS(einträge, id, top, value)  
Einfache Liste, ausgewählter Eintrag wird markiert

ListROC(einträge, id, top)  
Wie 'ListRO()', keine Steuerung mit den Cursortasten

ListSelC(einträge, id, top)  
Wie 'ListSel()', keine Steuerung mit den Cursortasten

ListSSC(einträge, id, top, value)  
Wie 'ListSS()', keine Steuerung mit den Cursortasten

ListRON(einträge, id, top)  
Wie 'ListRO()', keine Steuerung mit dem Zehnerblock

ListSelN(einträge, id, top)  
Wie 'ListSel()', keine Steuerung mit dem Zehnerblock

ListSSN(einträge, id, top, value)  
Wie 'ListSS()', keine Steuerung mit dem Zehnerblock

ListROCN(einträge, id, top)  
Wie 'ListRO()', keine Steuerung mit den Cursortasten und dem  
Zehnerblock

ListSelCN(einträge, id, top)  
Wie 'ListSel()', keine Steuerung mit den Cursortasten und dem  
Zehnerblock

ListSSCN(einträge, id, top, value)  
Wie 'ListSS()', keine Steuerung mit den Cursortasten und dem  
Zehnerblock

-----

FWListRO(einträge, id, top)  
Wie 'ListRO()', Zeichensatz mit fester Breite

FWListSel(einträge, id, top)  
Wie 'ListSel()', Zeichensatz mit fester Breite

FWListSS(einträge, id, top, value)  
Wie 'ListSS()', Zeichensatz mit fester Breite

---



```

FWListROC (einträge, id, top)
    Wie 'ListROC()', Zeichensatz mit fester Breite

FWListSelC (einträge, id, top)
    Wie 'ListSelC()', Zeichensatz mit fester Breite

FWListSSC (einträge, id, top, value)
    Wie 'ListSSC()', Zeichensatz mit fester Breite

FWListRON (einträge, id, top)
    Wie 'ListRON()', Zeichensatz mit fester Breite

FWListSelN (einträge, id, top)
    Wie 'ListSelN()', Zeichensatz mit fester Breite

FWListSSN (einträge, id, top, value)
    Wie 'ListSSN()', Zeichensatz mit fester Breite

FWListROCN (einträge, id, top)
    Wie 'ListROCN()', Zeichensatz mit fester Breite

FWListSelCN (einträge, id, top)
    Wie 'ListSelCN()', Zeichensatz mit fester Breite

FWListSSCN (einträge, id, top, value)
    Wie 'ListSSCN()', Zeichensatz mit fester Breite

```

## DEFINITIONEN

-/-

**1.25 TritonRexx.guide/Palette**

Palette

\*\*\*\*\*

## OBJEKTNAME

TROB\_Palette

## ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_Palette	Ungenutzt	
TRAT_Disabled	Bool	Palette anwählbar?
TRAT_ID	Integer	ID-Nummer der Palette
TRAT_Value	Integer	Nummer der ausgewählten Farbe
TRDO_QuickHelpString	Zeichenkette	Kurzer Hilfstext

## FUNKTIONEN

PaletteGadget (value, id)

Einfache Palette zur Farbauswahl

DEFINITIONEN

-/-

## 1.26 TritonRexx.guide/Scroller

Scroller

\*\*\*\*\*

OBJEKTNAME

TROB\_Scroller

ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_Scroller	Flags	
	TROF_HORIZ	Horizontaler Scroller
	TROF_VERT	Vertikaler Scroller
TRAT_Disabled	Bool	Scroller anwählbar?
TRAT_ID	Integer	ID-Nummer des Scrollers
TRAT_Value	Integer	Nummer des ersten sichtbaren Eintrags der Liste, die der Scroller steuert
TRSC_Total	Integer	Gesamtanzahl der Einträge der Liste, die der Scroller steuert
TRSC_Visible	Integer	Anzahl der sichtbaren Einträge der Liste, die der Regler steuert
TRDO_QuickHelpString	Zeichenkette	Kurzer Hilfstext

FUNKTIONEN

-/-

DEFINITIONEN

-/-

## 1.27 TritonRexx.guide/Slider

Slider

\*\*\*\*\*

OBJEKTNAME

TROB\_Slider

## ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_Slider	Flags	
	TROF_HORIZ	Horizontaler Schieberegler
	TROF_VERT	Vertikaler Schieberegler
TRAT_Disabled	Bool	Schieberegler anwählbar?
TRAT_ID	Integer	ID-Nummer des Schiebereglers
TRAT_Value	Integer	Anzuzeigender Wert
TRSL_Min	Integer	Kleinster einstellbarer Wert
TRSL_Max	Integer	Größter einstellbarer Wert
TRDO_QuickHelpString	Zeichenkette	Kurzer Hilfstext

## FUNKTIONEN

SliderGadget (minimum,maximum,value,id)  
Einfacher, horizontaler Schieberegler

## DEFINITIONEN

-/-

## WICHTIG

Mit Slider-Gadgets können in der aktuellen Version keine negativen Zahlen eingestellt werden. Falls sich die Verwendung von negativen Zahlen nicht vermeiden läßt kann man sich mit einem Trick behelfen. Man wählt einen Einstellbereich von '0 bis Maximalwert-Minimalwert', der angezeigte Wert errechnet sich dann aus 'SliderValue+Minimalwert'.

## Beispiel:

```
-----
minimum = -128
maximum = 127
wert     = -100
```

## Slider erstellen:

```
-----
SliderGadget (0,maximum-minimum,wert+minimum,1)
```

## Anzuzeigenden Wert ermitteln:

```
-----
wert = TR_GETATTRIBUTE (project,1,'TRAT_Value')+minimum
```

## Wert aus einer 'TRMS\_NEWVALUE-Nachricht' auslesen:

```
-----
wert = stemvar.trm_data+minimum
```

## 1.28 TritonRexx.guide/String

String

\*\*\*\*\*

OBJEKTNAME

TROB\_String

ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_String	Zeichenkette	Der zu edierende Text
TRAT_Disabled	Bool	Eingabefeld aktivierbar?
TRAT_Flags	Flags	
	TRST_INVISIBLE	Die Eingabe ist nicht sichtbar (Passwort)
	TRST_NORETURNBROADCAST	RETURN-Meldung nicht weiterreichen
TRAT_ID	Integer	ID-Nummer des Eingabefeldes
TRAT_Value	Integer	Maximale Anzahl Zeichen
TRDO_QuickHelpString	Zeichenkette	Kurzer Hilfstext

FUNKTIONEN

StringGadget (defaultstring, id)

Ein normales Texteingabefeld

PasswordGadget (defaultstring, id)

Ein Eingabefeld für Passwörter

DEFINITIONEN

-/-

## 1.29 TritonRexx.guide/FrameBox

FrameBox

\*\*\*\*\*

OBJEKTNAME

TROB\_FrameBox

ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_FrameBox	Typ	
	TRFB_TEXT	Rahmen für eine Textbox
	TRFB_GROUPING	Ein Gruppen-Rahmen

	TRFB_FRAMING	Einrahmung einer Gruppe
TRAT_Backfill	Füllmuster Füllmuster für Objekte .	Siehe
TRAT_ID	Integer	ID-Nummer des Rahmens
TRAT_Text	Zeichenkette	Text für den Rahmen

Es wird nur das nachfolgende Objekt eingerahmt. Wenn eine ganze Gruppe eingerahmt werden soll muß also das erste Objekt nach der FrameBox die einzurahmende Gruppe sein. Siehe  
Group  
.

#### FUNKTIONEN

NamedFrameBox(text)  
Einrahmung einer Gruppe mit Text als Überschrift

TextBox(text,id,minwidth)  
Eine Textbox mit mindestens 'minwidth' Zeichen Breite

#### DEFINITIONEN

GroupBox  
Einfacher Gruppenrahmen

## 1.30 TritonRexx.guide/Group

Group

\*\*\*\*\*

#### OBJEKTNAME

TRGR\_Horiz TRGR\_Vert

#### ATTRIBUTE

Kennung	Datenangabe	Beschreibung
---------	-------------	--------------

TRGR_Horiz	Flags Oberflächen erstellen .	Siehe
------------	-------------------------------------	-------

TRGR_Vert	Flags Oberflächen erstellen .	Siehe
-----------	-------------------------------------	-------

TRGR_PROPSHARE	Objekte proportional anordnen
TRGR_EQUALSHARE	Objekte gleichmäßig anordnen
TRGR_PROPSPACES	Leerräume skalieren
TRGR_ARRAY	Tabelle von Objekten
TRGR_ALIGN	Objekte aneinander anpassen
TRGR_CENTER	Objekte zueinander zentrieren
TRGR_FIXHORIZ	Gruppe mit fester Breite
TRGR_FIXVERT	Gruppe mit fester Höhe

---

---

	TRGR_INDEP	Gruppe ist von der übergeordneten Tabelle unabhängig
TRAT_ID	Integer	ID-Nummer der Gruppe
TRGR_End	Ungenutzt	Markiert das Ende einer Gruppe

## FUNKTIONEN

-/-

## DEFINITIONEN

## HorizGroup

Einfache horizontale Gruppe

## HorizGroupE

Horizontale Gruppe, alle Objekte haben die gleiche Breite

## HorizGroupS

Horizontale Gruppe, alle Objekte haben die minimale Breite,  
Leerräume werden skaliert

## HorizGroupA

Horizontale Gruppe, alle Objekte haben die gleiche Höhe

## HorizGroupC

Horizontale Gruppe, alle Objekte werden vertikal zentriert  
angeordnet

## HorizGroupEA

Kombination aus 'HorizGroupE' und 'HorizGroupA'

## HorizGroupSA

Kombination aus 'HorizGroupS' und 'HorizGroupA'

## HorizGroupEC

Kombination aus 'HorizGroupE' und 'HorizGroupC'

## HorizGroupSC

Kombination aus 'HorizGroupS' und 'HorizGroupC'

## HorizGroupAC

Kombination aus 'HorizGroupA' und 'HorizGroupC'

## HorizGroupEAC

Kombination aus 'HorizGroupE', 'HorizGroupA' und 'HorizGroupC'

## HorizGroupSAC

Kombination aus 'HorizGroupS', 'HorizGroupA' und 'HorizGroupC'

-----

## VertGroup

Einfache vertikale Gruppe

## VertGroupE

Vertikale Gruppe, alle Objekte haben die gleiche Höhe

VertGroupS  
Vertikale Gruppe, alle Objekte haben die minimale Höhe, Leerräume werden skaliert

VertGroupA  
Vertikale Gruppe, alle Objekte haben die gleiche Breite

VertGroupC  
Vertikale Gruppe, alle Objekte werden horizontal zentriert angeordnet

VertGroupEA  
Kombination aus 'VertGroupE' und 'VertGroupA'

VertGroupSA  
Kombination aus 'VertGroupS' und 'VertGroupA'

VertGroupEC  
Kombination aus 'VertGroupE' und 'VertGroupC'

VertGroupSC  
Kombination aus 'VertGroupS' und 'VertGroupC'

VertGroupAC  
Kombination aus 'VertGroupA' und 'VertGroupC'

VertGroupEAC  
Kombination aus 'VertGroupE', 'VertGroupA' und 'VertGroupC'

VertGroupSAC  
Kombination aus 'VertGroupS', 'VertGroupA' und 'VertGroupC'

-----  
EndGroup  
Ende einer Gruppe

ColumnArray  
Anfang einer Spalten-Tabelle

LineArray  
Anfang einer Zeilen-Tabelle

BeginColumn  
Anfang einer neuen Spalte innerhalb der Tabelle

BeginLine  
Anfang einer neuen Zeile innerhalb der Tabelle

BeginColumnI  
Anfang einer unabhängigen Spalte innerhalb der Tabelle

BeginLineI  
Anfang einer unabhängigen Zeile innerhalb der Tabelle

EndColumn  
Ende einer Spalte innerhalb der Tabelle

---

EndLine  
 Ende einer Zeile innerhalb der Tabelle

EndArray  
 Ende einer Tabelle

## 1.31 TritonRexx.guide/Image

Image

\*\*\*\*\*

OBJEKTNAME  
 TROB\_Image

ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_Image	Image	Der logische Name einer zuvor angelegten Grafik Siehe
	CreateImage()	.
TRAT_Flags	Flags	
	TRIM_BOOPSI	In der aktuellen Version muß dieses Flag immer angegeben werden, sonst kann es zu Abstürzen kommen!
TRAT_MinWidth	Integer	Minimale Breite der Grafik in Pixel (Standard ist die Höhe eines Knopfes)
TRAT_MinHeight	Integer	Minimale Höhe der Grafik in Pixel (Standard ist die Höhe eines Knopfes)
TRAT_ID	Integer	ID-Nummer der Grafik

FUNKTIONEN

BoopsiImage(image)  
 Einfache Grafik

BoopsiImageD(image,minwidth,minheight)  
 Einfache Grafik, jedoch mit minimaler Breite und Höhe

DEFINITIONEN

-/-



## 1.32 TritonRexx.guide/Line

Line

\*\*\*\*

OBJEKTNAME

TROB\_Line

ATTRIBUTE

Kennung	Datenangabe	Beschreibung
-----		
TROB_Line	Flags	
	TROF_HORIZ	Horizontale linie
	TROF_VERT	Vertikale linie
	TROF_RAISED	Linie ist erhaben

FUNKTIONEN

Line(flags)

Einfache Linie

NamedSeparator(text)

Horizontale Trennlinie mit Leerraum an Anfang und Ende und Text in der Mitte

NamedSeparatorI(te,id)

Wie 'NamedSeparator()', nur mit ID-Nummer

NamedSeparatorN(text)

Horizontale Trennlinie mit Text in der Mitte

NamedSeparatorIN(te,id)

Wie 'NamedSeparatorN()', nur mit ID-Nummer

DEFINITIONEN

HorizSeparator

Horizontale Trennlinie mit Leerraum an Anfang und Ende

VertSeparator

Vertikale Trennlinie mit Leerraum oben und unten

## 1.33 TritonRexx.guide/Progress

Progress

\*\*\*\*\*

OBJEKTNAME

TROB\_Progress

ATTRIBUTE

Kennung	Datenangabe	Beschreibung
-----		
TROB_Progress	Integer	Maximaler Wert,

z.B. 100 bei Prozentanzeige

TRAT_Flags	Flags	
	TROF_HORIZ	Horizontal (Standard)
	TROF_VERT	Vertikal
TRAT_ID	Integer	ID-Nummer
TRAT_Value	Integer	Anzuzeigender Wert

#### FUNKTIONEN

Progress (maximum, value, id)  
 Horizontale Fortschrittsanzeige

#### DEFINITIONEN

-/-

## 1.34 TritonRexx.guide/Space

Space

\*\*\*\*\*

#### OBJEKTNAME

TROB\_Space

#### ATTRIBUTE

Kennung	Datenangabe	Beschreibung
---------	-------------	--------------

TROB_Space	Typ	
	TRST_NONE	Kein Leerraum
	TRST_SMALL	Kleiner Leerraum
	TRST_NORMAL	Normaler Leerraum
	TRST_BIG	Großer Leerraum

#### FUNKTIONEN

-/-

#### DEFINITIONEN

SpaceB  
 Großer Leerraum

Space  
 Normaler Leerraum

SpaceS  
 Kleiner Leerraum

SpaceN  
 Kein Leerraum

## 1.35 TritonRexx.guide/Text

Text

\*\*\*\*

OBJEKTNAME

TROB\_Text

ATTRIBUTE

Kennung	Datenangabe	Beschreibung
TROB_Text	Ungenutzt	
TRAT_ID	Integer	ID-Nummer des Textes
		Wenn ein anderes Objekt die gleiche ID besitzt, und im Text einem Zeichen ein '_' vorangestellt ist, dann kann das zugehörige Objekt mit der entsprechenden Taste gesteuert werden.
TRAT_Flags	Flags	
	TRTX_NOUNDERSCORE	'_' wird nicht beachtet
	TRTX_HIGHLIGHT	Hervorgehobener Text
	TRTX_3D	Text mit Schatten
	TRTX_BOLD	Fett geschriebener Text
	TRTX_TITLE	Titeltext, Attribute wie mit 'Prefs/Triton' eingestellt
	TROF_RIGHTALIGN	Text ist rechtsbündig
	TRTX_CLIPPED	Im zugehörigen Bereich nicht mehr darstellbarer Text wird abgeschnitten
TRAT_MinWidth	Integer	Minimale Breite des Textes in Zeichen
TRAT_Text	Zeichenkette	Der anzuzeigende Text
TRAT_Value	Integer	Eine anzuzeigende Zahl

FUNKTIONEN

TextN(text)

Normaler Text

TextH(text)

Hervorgehobener Text

Text3(text)

Text mit Schatten

TextB(text)

Fett geschriebener Text

TextT(text)

Titeltext, Attribute wie mit 'Prefs/Triton' eingestellt

TextID(text,id)  
Normaler Text mit ID

TextNR(text)  
Normaler, rechtsbündiger Text

TextRIGHT(text,id)  
Rechtsbündiger Text mit ID

ClippedText(text)  
Normaler Text, zu langer Text wird abgeschnitten

ClippedTextID(text,id)  
Normaler Text mit ID, zu langer Text wird abgeschnitten

CenteredText(text)  
Zentrierter Text

CenteredTextH(text)  
Zentrierter, hervorgehobener Text

CenteredText3(text)  
Zentrierter Text mit Schatten

CenteredTextB(text)  
Zentrierter, fett geschriebener Text

CenteredTextID(text,id)  
Zentrierter Text mit ID

CenteredText\_BS(text)  
Zentrierter Text mit großem Leerraum an Anfang und Ende

-----

Integer(integer)  
Normale Integerzahl

IntegerH(integer)  
Hervorgehobene Integerzahl

Integer3(integer)  
Integerzahl mit Schatten

IntegerB(integer)  
Fett geschriebene Integerzahl

CenteredInteger(integer)  
Zentrierte Integerzahl

CenteredIntegerH(integer)  
Zentrierte, hervorgehobene Integerzahl

CenteredInteger3(integer)  
Zentrierte Integerzahl mit Schatten

CenteredIntegerB(integer)

---

Zentrierte, fett geschriebene Integerzahl

```

-----
TextBox(text,id,mwid)
    Ein Text-Anzeigefeld

ClippedTextBox(text,id)
    Wie 'TextBox()', zu langer Text wird abgeschnitten

ClippedTextBoxMW(text,id,minwidth)
    Wie 'ClippedTextBox()', mit Angabe der minimalen Breite

IntegerBox(integer,id,minwidth)
    Ein Anzeigefeld für Integerzahlen
  
```

#### DEFINITIONEN

-/-

## 1.36 TritonRexx.guide/QuickHelp

QuickHelp

\*\*\*\*\*

#### BESCHREIBUNG

Vielen Objekten kann ein kurzer Hilfstext zugeordnet werden, der in einem kleinen Fenster angezeigt wird wenn sich der Mauszeiger über dem zugehörigen Objekt befindet.

#### FUNKTIONEN

QuickHelp(helpstring)

Dies ist ein Makro, das bei allen Objekten verwendet werden kann

## 1.37 TritonRexx.guide/Füllmuster für Objekte

Füllmuster für Objekte

\*\*\*\*\*

#### BESCHREIBUNG

Kennung	Datenangabe	Beschreibung
---------	-------------	--------------

----- ↩

TRAT_Backfill	Füllmuster	
	TRBF_WINDOWBACK	Fenster-Hintergrund
	TRBF_REQUESTERBACK	Requester-Hintergrund
	TRBF_NONE	Hintergrund
	TRBF_SHINE	Farbe der hellen Kanten
	TRBF_SHINE_SHADOW	Farben der hellen und dunklen Kanten
	TRBF_SHINE_FILL	Farbe der hellen Kanten und Füllfarbe

TRBF_SHINE_BACKGROUND	Farbe der hellen Kanten und Hintergrund	↔
TRBF_SHADOW	Farbe der dunklen Kanten	
TRBF_SHADOW_FILL	Farbe der dunklen Kanten und Füllfarbe	
TRBF_SHADOW_BACKGROUND	Farbe der dunklen Kanten und Hintergrund	↔
TRBF_FILL	Füllfarbe	
TRBF_FILL_BACKGROUND	Füllfarbe und Hintergrund	

## DEFINITIONEN

ObjectBackfillWin

Fenster-Hintergrund

ObjectBackfillReq

Requester-Hintergrund

ObjectBackfillB

Screen-Hintergrund

ObjectBackfills

Farbe der hellen Kanten

ObjectBackfillsA

Farben der hellen und dunklen Kanten

ObjectBackfillsF

Farbe der hellen Kanten und Füllfarbe

ObjectBackfillsB

Farbe der hellen Kanten und Hintergrundfarbe

ObjectBackfillsA

Farbe der dunklen Kanten

ObjectBackfillsAF

Farbe der dunklen Kanten und Füllfarbe

ObjectBackfillsAB

Farbe der dunklen Kanten und Hintergrundfarbe

ObjectBackfillsF

Füllfarbe

ObjectBackfillsFB

Füllfarbe und Hintergrundfarbe

## 1.38 TritonRexx.guide/Füllmuster für Fenster

Füllmuster für Fenster

\*\*\*\*\*

## BESCHREIBUNG

Kennung

Datenangabe

Beschreibung

----- ←

TRWI\_Backfill Füllmuster

TRBF_WINDOWBACK	Fenster-Hintergrund
TRBF_REQUESTERBACK	Requester-Hintergrund
TRBF_NONE	Hintergrund
TRBF_SHINE	Farbe der hellen Kanten
TRBF_SHINE_SHADOW	Farben der hellen und dunklen Kanten
TRBF_SHINE_FILL	Farbe der hellen Kanten und Füllfarbe
TRBF_SHINE_BACKGROUND	Farbe der hellen Kanten und ←
	Hintergrund
TRBF_SHADOW	Farbe der dunklen Kanten
TRBF_SHADOW_FILL	Farbe der dunklen Kanten und Füllfarbe
TRBF_SHADOW_BACKGROUND	Farbe der dunklen Kanten und ←
	Hintergrund
TRBF_FILL	Füllfarbe
TRBF_FILL_BACKGROUND	Füllfarbe und Hintergrund

#### DEFINITIONEN

WindowBackfillWin

Fenster-Hintergrund

WindowBackfillReq

Requester-Hintergrund

WindowBackfillNone

Screen-Hintergrund

WindowBackfills

Farbe der hellen Kanten

WindowBackfillsA

Farben der hellen und dunklen Kanten

WindowBackfillsF

Farbe der hellen Kanten und Füllfarbe

WindowBackfillsB

Farbe der hellen Kanten und Hintergrundfarbe

WindowBackfillsA

Farbe der dunklen Kanten

WindowBackfillsAF

Farbe der dunklen Kanten und Füllfarbe

WindowBackfillsAB

Farbe der dunklen Kanten und Hintergrundfarbe

WindowBackfillsF

Füllfarbe

WindowBackfillsFB

Füllfarbe und Hintergrundfarbe

## 1.39 TritonRexx.guide/Funktionen

Funktionen der Library

\*\*\*\*\*

Alle Funktionen können auch über ein Alias aufgerufen werden, indem man dem Funktionsnamen ein 'TRX\_' voranstellt. Siehe  
Tips und Problemlösungen  
.

Triton-Funktionen

TR\_CreateApp()  
    Applikation erzeugen

TR\_DeleteApp()  
    Applikation löschen

TR\_OpenProject()  
    Projekt/Fenster öffnen

TR\_CloseProject()  
    Projekt/Fenster schließen

TR\_LockProject()  
    Projekt/Fenster für Eingaben sperren

TR\_UnlockProject()  
    Projekt/Fenster für Eingaben freigeben

TR\_LockScreen()  
    Bildschirm eines Projekts übernehmen

TR\_UnlockScreen()  
    Bildschirm eines Projekts freigeben

TR\_ObtainWindow()  
    Fenster eines Projekts übernehmen

TR\_ReleaseWindow()  
    Fenster eines Projekts freigeben

TR\_AutoRequest()  
    Fenster als Requester öffnen

TR\_EasyRequest()  
    Einfachen Requester öffnen

TR\_GetAttribute()  
    Attribut eines Objekts abfragen

---



TR\_SetAttribute()  
Attribut eines Objekts ändern

TR\_SendMessage()  
Nachricht an ein Objekt schicken

TR\_Wait()  
Auf Nachrichten einer Applikation warten

TR\_GetMsg()  
Nachricht von einer Applikation empfangen

TR\_ReplyMsg()  
Nachricht einer Applikation beantworten

TR\_HandleMsg()  
Nachricht empfangen und beantworten

TR\_FirstOccurance()  
Zeichen in einem Text suchen

TR\_NumOccurances()  
Zeichen in einem Text zählen

TR\_GetLastError()  
Zuletzt aufgetretenen Fehler ermitteln

TR\_GetErrorString()  
Fehlerbeschreibung anfordern

#### Datei- und Fontrequester

ASL\_RequestFile()  
Datei-Requester aufrufen

ASL\_RequestFont()  
Font-Requester aufrufen

#### Grafiken

CreateImage()  
Grafik von Disk laden

DeleteImage()  
Grafik aus dem Speicher löschen

ImageWidth()  
Breite einer geladenen Grafik ermitteln

ImageHeight()  
Höhe einer geladenen Grafik ermitteln

---

Sonstiges

```
ActivateProject()
    Fenster eines Projekts aktivieren

ProjectToFront()
    Fenster eines Projekts nach vorne

ProjectToBack()
    Fenster eines Projekts nach hinten

TritonRexxVersion()
    Version der 'tritonrexx.library' ermitteln

TritonVersion()
    Version der 'triton.library' ermitteln
```

## 1.40 TritonRexx.guide/TR\_CreateApp()

Die Funktion TR\_CreateApp() - Applikation erzeugen

\*\*\*\*\*

NAME

```
app = TR_CreateApp(taglist)
```

BESCHREIBUNG

Erzeugt eine Applikation. Eine Applikation ist die Grundlage für alle Fenster, daher muß diese Funktion immer aufgerufen werden bevor Fenster geöffnet werden können.

ARGUMENTE

taglist - Eine Tagliste, die die Beschreibung der Applikation enthält.  
Siehe  
Applikationen  
.

ERGEBNIS

app - Eine Kennzahl für die erzeugte Applikation im binären Format.  
'00000000'x wenn die Applikation nicht erzeugt werden konnte.

SIEHE AUCH

```
TR_DeleteApp()
    Applikation löschen
```

## 1.41 TritonRexx.guide/TR\_DeleteApp()

Die Funktion TR\_DeleteApp() - Applikation löschen

\*\*\*\*\*

### NAME

dummy = TR\_DeleteApp(app)

### BESCHREIBUNG

Löscht eine zuvor mit 'TR\_CreateApp()' erzeugte Applikation. Alle zu dieser Applikation gehörenden Projekte/Fenster werden ebenfalls geschlossen!

### ARGUMENTE

app - Die Kennzahl der Applikation, die gelöscht werden soll  
Dies ist der Rückgabewert von 'TR\_CreateApp()'.

### ERGEBNIS

dummy - immer 1

### SIEHE AUCH

TR\_CreateApp()  
Applikation erzeugen

## 1.42 TritonRexx.guide/TR\_OpenProject()

Die Funktion TR\_OpenProject() - Projekt/Fenster öffnen

\*\*\*\*\*

### NAME

project = TR\_OpenProject(app,taglist)

### BESCHREIBUNG

Öffnet ein Projekt/Fenster

### ARGUMENTE

app - Die Kennzahl der Applikation, zu der das Fenster gehört.  
Dies ist der Rückgabewert von 'TR\_CreateApp()'.

taglist - Tagliste, die die Beschreibung des Projekts enthält. Die Eigenschaften müssen in der Tagliste in einer bestimmten Reihenfolge angegeben werden:

1. Fensterbeschreibung     Siehe  
   Projekte-Fenster
- .
2. Menübeschreibungen     Siehe  
   Menüs
- .
3. Objektbeschreibungen   Siehe

Objekte und Attribute

.

ERGEBNIS

project - Eine Kennzahl für das geöffnete Projekt im binären Format.  
'00000000'x wenn das Projekt nicht geöffnet werden konnte.

SIEHE AUCH

TR\_CreateApp()  
Applikation erzeugen

TR\_CloseProject()  
Projekt/Fenster schließen

### 1.43 TritonRexx.guide/TR\_CloseProject()

Die Funktion TR\_CloseProject() - Projekt/Fenster schließen

\*\*\*\*\*

NAME

dummy = TR\_CloseProject(project)

BESCHREIBUNG

Schließt ein zuvor mit 'TR\_OpenProject()' geöffnetes Projekt.

ARGUMENTE

project - Die Kennzahl des Projekts, das gelöscht werden soll.  
Dies ist der Rückgabewert von 'TR\_OpenProject()'.

ERGEBNIS

dummy - immer 1

SIEHE AUCH

TR\_OpenProject()  
Projekt/Fenster öffnen

### 1.44 TritonRexx.guide/TR\_LockProject()

Die Funktion TR\_LockProject() - Projekt/Fenster für Eingaben ↔  
sperrern

\*\*\*\*\*

NAME

dummy = TR\_LockProject(project)

## BESCHREIBUNG

Sperrt ein Projekt. Im Fenster des gesperrten Projekts sind keinerlei Eingaben mehr möglich. Das Verkleinern und Vergrößern des Fensters ist jedoch weiterhin möglich.

## ARGUMENTE

project - Die Kennzahl des Projekts, das für Eingaben gesperrt werden soll. Dies ist der Rückgabewert von 'TR\_OpenProject()'.

## ERGEBNIS

dummy - immer 1

## SIEHE AUCH

TR\_OpenProject()  
Projekt/Fenster öffnen

TR\_UnlockProject()  
Projekt/Fenster für Eingaben freigeben

## 1.45 TritonRexx.guide/TR\_UnlockProject()

Die Funktion TR\_UnlockProject() - Projekt/Fenster für Eingaben ↔ freigeben

\*\*\*\*\*

## NAME

dummy = TR\_UnlockProject(project)

## BESCHREIBUNG

Gibt ein zuvor gesperrtes Projekt wieder für Eingaben frei.

## ARGUMENTE

project - Die Kennzahl des Projekts, das freigegeben werden soll. Dies ist der gleiche Wert, der zuvor bei 'TR\_LockProject()' angegeben wurde.

## ERGEBNIS

dummy - immer 1

## SIEHE AUCH

TR\_LockProject()  
Projekt/Fenster für Eingaben sperren

## 1.46 TritonRexx.guide/TR\_LockScreen()

Die Funktion TR\_LockScreen() - Bildschirm eines Projekts ↔  
übernehmen

\*\*\*\*\*

### NAME

screen = TR\_LockScreen(project)

### BESCHREIBUNG

Liefert einen Zeiger auf den Bildschirm des angegebenen Projekts, und verhindert, daß der Bildschirm geschlossen werden kann. Der Bildschirm kann jetzt gefahrlos zum Öffnen von anderen Fenstern verwendet werden.

### ARGUMENTE

project - Die Kennzahl des Projekts, dessen Bildschirm übernommen werden soll. Dies ist der Rückgabewert von 'TR\_OpenProject()'.  
soll.

### ERGEBNIS

screen - Zeiger auf die Screen-Struktur des Bildschirms im binären Format, '00000000'x im Fehlerfall.

### SIEHE AUCH

TR\_OpenProject()  
Projekt/Fenster öffnen

TR\_UnlockScreen()  
Bildschirm eines Projekts freigeben

## 1.47 TritonRexx.guide/TR\_UnlockScreen()

Die Funktion TR\_UnlockScreen() - Bildschirm eines Projekts ↔  
freigeben

\*\*\*\*\*

### NAME

dummy = TR\_UnlockScreen(screen)

### BESCHREIBUNG

Gibt einen zuvor mit 'TR\_LockScreen()' übernommenen Bildschirm frei. Der Bildschirm kann erst nach dem Aufruf dieser Funktion wieder geschlossen werden.

### ARGUMENTE

screen - Zeiger auf die Screen-Struktur des Bildschirms, der freigegeben werden soll. Dies ist der Rückgabewert von 'TR\_LockScreen()'.  
werden soll.

### ERGEBNIS

dummy - immer 1

SIEHE AUCH

```
TR_LockScreen()  
    Bildschirm eines Projekts übernehmen
```

## 1.48 TritonRexx.guide/TR\_ObtainWindow()

Die Funktion TR\_ObtainWindow() - Fenster eines Projekts übernehmen  
\*\*\*\*\*

NAME

```
window = TR_ObtainWindow(project)
```

BESCHREIBUNG

Liefert einen Zeiger auf das Fenster des angegebenen Projekts, und erlaubt es das Fenster mit 'Nicht-Triton-Funktionen' zu manipulieren. Man kann z.B. das Fenster nach vorne oder hinten legen, aktivieren oder deaktivieren. Der Inhalt des Fensters darf auf keinen Fall verändert werden! Das Fenster muß später mit 'TR\_ReleaseWindow()' wieder freigegeben werden.

ARGUMENTE

project - Die Kennzahl des Projekts, dessen Fenster übernommen werden soll. Dies ist der Rückgabewert von 'TR\_OpenProject()'.

ERGEBNIS

window - Zeiger auf die Window-Struktur des Fensters im binären Format, '00000000'x im Fehlerfall.

SIEHE AUCH

```
TR_OpenProject()  
    Projekt/Fenster öffnen  
  
TR_ReleaseWindow()  
    Fenster eines Projekts freigeben  
  
ProjectToFront()  
    Fenster eines Projekts nach vorne  
  
ProjectToBack()  
    Fenster eines Projekts nach hinten
```

## 1.49 TritonRexx.guide/TR\_ReleaseWindow()

Die Funktion TR\_ReleaseWindow() - Fenster eines Projekts freigeben

\*\*\*\*\*

NAME

dummy = TR\_ReleaseWindow(window)

BESCHREIBUNG

Gibt ein zuvor mit 'TR\_ObtainWindow()' übernommenes Fenster frei.

ARGUMENTE

window - Zeiger auf die Window-Struktur des Fensters, das freigegeben werden soll. Dies ist der Rückgabewert von 'TR\_ObtainWindow()'.

ERGEBNIS

dummy - immer 1

SIEHE AUCH

TR\_ObtainWindow()  
Fenster eines Projekts übernehmen

## 1.50 TritonRexx.guide/ActivateProject()

Die Funktion ActivateProject() - Fenster eines Projekts aktivieren

\*\*\*\*\*

NAME

dummy = ActivateProject(project)

BESCHREIBUNG

Aktiviert das zum angegebenen Projekt gehörende Fenster.

ARGUMENTE

project - Die Kennzahl des Projekts, dessen Fenster aktiviert werden soll, Rückgabewert von 'TR\_OpenProject()'.

ERGEBNIS

dummy - immer 1

SIEHE AUCH

ProjectToBack()  
Fenster eines Projekts nach hinten

ProjectToFront()  
Fenster eines Projekts nach hinten

TR\_ObtainWindow()  
Fenster eines Projekts übernehmen

---



```
TR_ReleaseWindow()  
    Fenster eines Projekts freigeben
```

## 1.51 TritonRexx.guide/ProjectToFront()

```
Die Funktion ProjectToFront() - Fenster eines Projekts nach vorne  
*****
```

### NAME

```
dummy = ProjectToFront(project)
```

### BESCHREIBUNG

Bringt das zum angegebenen Projekt gehörende Fenster in den Vordergrund.

### ARGUMENTE

project - Die Kennzahl des Projekts, dessen Fenster in den Vordergrund gebracht werden soll, Rückgabewert von 'TR\_OpenProject()'.

### ERGEBNIS

```
dummy - immer 1
```

### SIEHE AUCH

```
ActivateProject()  
    Fenster eines Projekts aktivieren
```

```
ProjectToBack()  
    Fenster eines Projekts nach hinten
```

```
TR_ObtainWindow()  
    Fenster eines Projekts übernehmen
```

```
TR_ReleaseWindow()  
    Fenster eines Projekts freigeben
```

## 1.52 TritonRexx.guide/ProjectToBack()

```
Die Funktion ProjectToBack() - Fenster eines Projekts nach hinten  
*****
```

### NAME

```
dummy = ProjectToBack(project)
```

### BESCHREIBUNG

Bringt das zum angegebenen Projekt gehörende Fenster in den Hintergrund.

### ARGUMENTE

project - Die Kennzahl des Projekts, dessen Fenster in den Hintergrund gebracht werden soll, Rückgabewert von 'TR\_OpenProject()'.

#### ERGEBNIS

dummy - immer 1

#### SIEHE AUCH

ActivateProject()  
Fenster eines Projekts aktivieren

ProjectToFront()  
Fenster eines Projekts nach vorne

TR\_ObtainWindow()  
Fenster eines Projekts übernehmen

TR\_ReleaseWindow()  
Fenster eines Projekts freigeben

## 1.53 TritonRexx.guide/TR\_AutoRequest()

Die Funktion TR\_AutoRequest() - Fenster als Requester öffnen

\*\*\*\*\*

#### NAME

auswahl = TR\_AutoRequest(app,project,taglist)

#### BESCHREIBUNG

Öffnet einen durch die Tagliste beschriebenen Requester. Sobald eine Nachricht vom Typ 'TRMS\_ACTION' eintrifft wird der Requester sofort wieder geschlossen. Wenn ein Projekt angegeben ist, wird dieses in der Zeit, in der der Requester geöffnet ist, für Eingaben gesperrt.

#### ARGUMENTE

app - Die Kennzahl der Applikation, zu der der Requester gehört. Dies ist der Rückgabewert von 'TR\_CreateApp()'.

project - Die Kennzahl des Projekts, das für Eingaben gesperrt werden soll. Dies ist der Rückgabewert von 'TR\_OpenProject()'. '00000000'x wenn kein Projekt gesperrt werden soll.

taglist - Tagliste, die die Beschreibung des Requesters enthält. Siehe TR\_OpenProject()

.

#### ERGEBNIS

auswahl - Die ID-Nummer des Objekts, das die 'TRMS\_ACTION'-Nachricht ausgelöst hat. -1 wenn das Schließsymbol des Fensters angeklickt wurde.

SIEHE AUCH

TR\_EasyRequest()  
Einfachen Requester öffnen

## 1.54 TritonRexx.guide/TR\_EasyRequest()

Die Funktion TR\_EasyRequest() - Einfachen Requester öffnen

\*\*\*\*\*

NAME

auswahl = TR\_EasyRequest(app,text,gadgets,taglist)

BESCHREIBUNG

Öffnet einen einfachen Requester.

ARGUMENTE

- app - Die Kennzahl der Applikation, zu der der Requester gehört.  
Dies ist der Rückgabewert von 'TR\_CreateApp()'.
- text - Text, der im Requester dargestellt werden soll.
- gadgets - Die Texte in den Gadgets des Requesters, getrennt durch '|'
- taglist - Eine Tagliste, die die Eigenschaften des Requesters enthält.

Kennung	Datenangabe	Beschreibung
TREZ_ReqPos	Position	Die Position des Requesters auf dem Bildschirm, hier gilt das gleiche wie beim Öffnen eines Projekts/Fensters. Standard ist 'TRWP_MOUSEPOINTER'. Siehe Projekte-Fenster .
TREZ_LockProject	Projekt	Die von 'TR_OpenProject()' erhaltene Kennzahl eines Projekts, das solange der Requester geöffnet ist für Eingaben gesperrt werden soll. Der Requester wird auf dem Bildschirm des hier angegebenen Projekts geöffnet.

Achtung:

-----  
Die Projekt-Kennzahl liegt im binären Format vor! Sie kann hier nicht einfach angegeben werden, sondern muß vorher in eine Zahl umgewandelt werden! Dazu wird die

		ARexx-Funktion 'C2D()' verwendet!
TREZ_Return	Integer	Die Nummer des Gadgets, das mit RETURN angewählt werden kann. Standard ist 1, bei Requestern mit nur einem Gadget 0.
TREZ_Title	Zeichenkette	Der Fenstertitel des Requesters. Standard ist 'System-Request'.
TREZ_Activate	Bool	Gibt an ob der Requester aktiviert ist. Wenn dies nicht angegeben ist wird der Zustand des Requesters von dem bei 'TREZ_LockProject' angegebenen Projekt übernommen.

#### ERGEBNIS

auswahl - Die Nummer des angewählten Gadgets, -1 wenn ein Fehler aufgetreten ist. Die Gadgets werden von links nach rechts durchnummeriert, beginnend bei 1. Das ganz rechte Gadget hat die Nummer 0.

#### SIEHE AUCH

TR\_AutoRequest()  
Fenster als Requester öffnen

## 1.55 TritonRexx.guide/TR\_GetAttribute()

Die Funktion TR\_GetAttribute() - Attribut eines Objekts abfragen

\*\*\*\*\*

#### NAME

value = TR\_GetAttribute(project,id,attribute)

#### BESCHREIBUNG

Eigenschaft eines Objekts oder Projekts abfragen. Eigenschaften von Objekten können nur abgefragt werden wenn das Objekt eine ID-Nummer besitzt.

#### ARGUMENTE

project - Die Kennzahl des Projekts, welches das abzufragende Objekt enthält. Dies ist der Rückgabewert von 'TR\_OpenProject()'.

id - ID-Nummer des Objekts, das abgefragt werden soll. Wenn eine Eigenschaft des Projekts abgefragt werden soll muß hier 0 angegeben werden.

attribute - Die Kennung der Eigenschaft, die abgefragt werden soll. Wenn die Standard-Eigenschaft eines Objekts abgefragt werden soll muß hier die Kennung des Objekts selbst angegeben werden! z.B. 'TROB\_String', wenn der in einem Eingabefeld enthaltene

Text abgefragt werden soll.

#### ERGEBNIS

value - Der abgefragte Wert. Dieser Wert hat je nach Objekt und Eigenschaft ein ganz bestimmtes Format. Es gilt immer das, was auch beim Öffnen eines Projekts angegeben wird.

Siehe  
 Taglisten verwenden  
 .  
 Siehe  
 Projekte-Fenster  
 .  
 Siehe  
 Menüs  
 .  
 Siehe  
 Objekte und Attribute  
 .

#### SIEHE AUCH

TR\_SetAttribute()  
 Attribut eines Objekts ändern

## 1.56 TritonRexx.guide/TR\_SetAttribute()

Die Funktion TR\_SetAttribute() - Attribut eines Objekts ändern

\*\*\*\*\*

#### NAME

dummy = TR\_SetAttribute(project,id,attribute,value)

#### BESCHREIBUNG

Eigenschaft eines Objekts oder Projekts ändern. Eigenschaften von Objekten können nur geändert werden wenn das Objekt eine ID-Nummer besitzt.

#### ARGUMENTE

project - Die Kennzahl des Projekts, welches das zu ändernde Objekt enthält. Dies ist der Rückgabewert von 'TR\_OpenProject()'.  
 id - ID-Nummer des Objekts, das geändert werden soll. Wenn eine Eigenschaft des Projekts geändert werden soll muß hier 0 angegeben werden.  
 attribute - Die Kennung der Eigenschaft, die geändert werden soll. Wenn die Standard-Eigenschaft eines Objekts geändert werden soll muß hier die Kennung des Objekts selbst angegeben werden! z.B. 'TROB\_String', wenn ein neuer Text in einem Eingabefeld angezeigt werden soll.

value - Der Wert, der neu eingestellt werden soll. Dieser Wert muß je nach Objekt und Eigenschaft ein ganz bestimmtes Format haben. Es gilt immer das, was auch beim Öffnen eines Projekts angegeben wird.

Siehe  
Taglisten verwenden  
.  
Siehe  
Projekte-Fenster  
.  
Siehe  
Menüs  
.  
Siehe  
Objekte und Attribute  
.

#### ERGEBNIS

dummy - immer 1

#### SIEHE AUCH

TR\_GetAttribute()  
Attribut eines Objekts abfragen

## 1.57 TritonRexx.guide/TR\_SendMessage()

Die Funktion TR\_SendMessage() - Nachricht an ein Objekt schicken  
\*\*\*\*\*

#### NAME

dummy = TR\_SendMessage(project,id,message,data)

#### BESCHREIBUNG

Sendet eine Nachricht an ein Objekt oder Projekt.

#### ARGUMENTE

project - Die Kennzahl des Projekts, welches das Objekt enthält.  
Dies ist der Rückgabewert von 'TR\_OpenProject()'.  
  
id - ID-Nummer des Objekts, an das die Nachricht geschickt werden soll. Soll die Nachricht an das Projekt geschickt werden, dann muß hier 0 angegeben werden.  
  
message - Die Kennung der Nachricht, die verschickt werden soll.  
  
data - Die Daten, die mit der Nachricht übermittelt werden sollen.

message	data	Beschreibung
TROM_ACTIVATE	Ungenutzt	Aktiviert ein Objekt

z.B.: Aktivieren eines Texteingabefeldes mit der ID 7

```
CALL TR_SendMessage(project,7,'TROM_ACTIVATE',0)
```

ERGEBNIS

dummy - immer 1

SIEHE AUCH

-/-

## 1.58 TritonRexx.guide/TR\_GetMsg()

Die Funktion TR\_GetMsg() - Nachricht von einer Applikation ←  
empfangen

\*\*\*\*\*

NAME

message = TR\_GetMsg(app,stemvar)

BESCHREIBUNG

Empfängt eine Nachricht von eine Applikation. Jede erhaltene Nachricht muß unbedingt mit der Funktion 'TR\_ReplyMsg()' beantwortet werden!

ARGUMENTE

app - Die Kennzahl der Applikation, von der eine Nachricht empfangen werden soll. Dies ist der Rückgabewert von 'TR\_CreateApp()'.

stemvar - Eine Stem-Variable, in der die Daten der Nachricht gespeichert werden sollen. Die Daten werden wie folgt gespeichert:

stemvar.trm\_project

Die Kennzahl des zur Nachricht gehörenden Projekts im binären Format

stemvar.trm\_id

Die ID-Nummer des Objekts, das die Nachricht ausgelöst hat

stemvar.trm\_class

Die Art der Nachricht:

TRMS\_CLOSEWINDOW - Das Fenster soll geschlossen werden, Schließsymbol wurde angeklickt oder ESC wurde gedrückt.

TRMS\_ERROR - Ein Fehler ist aufgetreten, der Fehlercode steht in 'stemvar.trm\_data'.

TRMS\_NEWVALUE - Daten eines Objekt wurden verändert, der neue Wert steht in 'stemvar.trm\_data'. z.B. Neuer Text im Eingabefeld oder der gerade ausgewählte Eintrag in einem Listview-Objekt.

- TRMS\_ACTION - Objekt löst eine Aktion aus, z.B. wenn ein Knopf angewählt wurde.
- TRMS\_ICONDROPPED - Ein oder mehrere Icons wurden auf dem Fenster, dann ist 'stemvar.trm\_id' = 0, oder einem 'DropBox'-Objekt, dann ist 'stemvar.trm\_id' die ID-Nummer, fallen gelassen. Die Pfadnamen der Icons stehen in 'stemvar.trm\_data':
- stemvar.trm\_data.0 - Anzahl der Icons  
stemvar.trm\_data.1 - erster Pfadname  
stemvar.trm\_data.2 - zweiter Pfadname  
stemvar.trm\_data.3 - dritter Pfadname  
stemvar.trm\_data.n - n'ter Pfadname
- TRMS\_KEYPRESSED - Eine Taste wurde gedrückt, die Tastencodes werden wie folgt gespeichert:
- stemvar.trm\_data - ASCII-Code  
stemvar.trm\_code - Tastatur-Code  
stemvar.trm\_qualifier - Qualifier-Code
- TRMS\_HELP - Der User fordert Hilfe für das Objekt mit der in 'stemvar.trm\_id' gespeicherten ID-Nummer an.
- TRMS\_DISKINSERTED - Eine Diskette wurde eingelegt.
- TRMS\_DISKREMOVED - Eine Diskette wurde entnommen.

stemvar.trm\_data  
Daten, je nach Art der Nachricht

stemvar.trm\_code  
Tastaturcode bei 'TRMS\_KEYPRESSED'

stemvar.trm\_qualifier  
Qualifiercode

stemvar.trm\_seconds  
Aktuelle Systemzeit in Sekunden seit dem 01.01.1978

stemvar.trm\_micros  
Microsekunden-Wert der Systemzeit

stemvar.trm\_app  
Die Kennzahl der zur Nachricht gehörenden Applikation im binären Format.

#### ERGEBNIS

message - Eine Kennzahl für die empfangene Nachricht im binären Format.  
Diese Kennzahl muß beim Beantworten der Nachricht mittels  
'TR\_ReplyMsg()' wieder angegeben werden! Alle Nachrichten  
müssen so schnell wie möglich beantwortet werden, um eine  
reibungslose Funktion zu gewährleisten!



SIEHE AUCH

```

TR_Wait()
    Auf Nachrichten einer Applikation warten

TR_ReplyMsg()
    Nachricht einer Applikation beantworten

TR_HandleMsg()
    Nachricht empfangen und beantworten

```

## 1.59 TritonRexx.guide/TR\_ReplyMsg()

Die Funktion TR\_ReplyMsg() - Nachricht einer Applikation ↵  
beantworten

\*\*\*\*\*

NAME

```
dummy = TR_ReplyMsg(message)
```

BESCHREIBUNG

Beantwortet eine Nachricht, die mit 'TR\_GetMsg()' empfangen wurde.

ARGUMENTE

message - Die Kennzahl der Nachricht, die beantwortet werden soll.  
Dies ist der Rückgabewert von 'TR\_GetMsg()'.

ERGEBNIS

```
dummy - immer 1
```

SIEHE AUCH

```

TR_Wait()
    Auf Nachrichten einer Applikation warten

TR_GetMsg()
    Nachricht von einer Applikation empfangen

TR_HandleMsg()
    Nachricht empfangen und beantworten

```

## 1.60 TritonRexx.guide/TR\_Wait()

Die Funktion TR\_Wait() - Nachricht einer Applikation beantworten

\*\*\*\*\*

## NAME

```
signale = TR_Wait(app,signale)
```

## BESCHREIBUNG

Wartet bis die angegebene Applikation eine Nachricht bereitstellt, oder auf das Auftreten von einem der angegebenen Unterbrechungs-Signale.

## ARGUMENTE

app - Die Kennzahl der Applikation, auf die gewartet werden soll.

signale - Hier können Signale angegeben werden, auf die zusätzlich gewartet werden soll. Es wird ein Text angegeben, der die folgenden Buchstaben enthalten kann:

C - es wird zusätzlich auf CTRL-C gewartet

D - es wird zusätzlich auf CTRL-D gewartet

E - es wird zusätzlich auf CTRL-E gewartet

F - es wird zusätzlich auf CTRL-F gewartet

z.B. 'CF' - wartet auf CTRL-C und CTRL-F

'' - wartet auf keine zusätzlichen Signale

## ERGEBNIS

signale - Die aufgetretenen Signale im oben beschriebenen Format.

## SIEHE AUCH

```
TR_GetMsg()
```

Nachricht von einer Applikation empfangen

```
TR_ReplyMsg()
```

Nachricht einer Applikation beantworten

```
TR_HandleMsg()
```

Nachricht empfangen und beantworten

## 1.61 TritonRexx.guide/TR\_HandleMsg()

Die Funktion TR\_HandleMsg() - Nachricht einer Applikation empfangen und beantworten ←

\*\*\*\*\* ←

## NAME

```
bool = TR_HandleMsg(app,stemvar)
```

## BESCHREIBUNG

Erfüllt den gleichen Zweck wie die Funktionen 'TR\_GetMsg()' und 'TR\_ReplyMsg()' zusammen, die empfangene Nachricht wird automatisch beantwortet!

## ARGUMENTE

app - Die Kennzahl der Applikation, von der eine Nachricht empfangen werden soll. Dies ist der Rückgabewert von 'TR\_CreateApp()'.

stemvar - Stem-Variable, in der die Daten der Nachricht gespeichert werden sollen. Siehe  
TR\_GetMsg()  
.

#### ERGEBNIS

bool - 1 wenn eine korrekte Nachricht empfangen wurde, 0 wenn nicht

#### SIEHE AUCH

TR\_Wait()

Auf Nachrichten einer Applikation warten

TR\_GetMsg()

Nachrichten von einer Applikation empfangen

TR\_ReplyMsg()

Nachricht einer Applikation beantworten

## 1.62 TritonRexx.guide/TR\_FirstOccurance()

Die Funktion TR\_FirstOccurance() - Zeichen in einem Text suchen

\*\*\*\*\*

#### NAME

position = TR\_FirstOccurance(char, string)

#### BESCHREIBUNG

Sucht in einer Zeichenkette nach dem ersten Auftreten eines Zeichens.

#### ARGUMENTE

char - Das Zeichen, nach dem gesucht werden soll.

string - Die Zeichenkette, die durchsucht werden soll.

#### ERGEBNIS

position - Die Position, an der das Zeichen zum ersten Mal auftritt.  
Das erste Zeichen hat die Nummer 1. 0 bedeutet, daß das Zeichen nicht in der Zeichenkette vorkommt.

#### SIEHE AUCH

TR\_NumOccurances()

Zeichen in einem Text zählen

## 1.63 TritonRexx.guide/TR\_NumOccurances()

Die Funktion TR\_NumOccurances() - Zeichen in einem Text zählen

\*\*\*\*\*

### NAME

count = TR\_NumOccurances(char, string)

### BESCHREIBUNG

Zählt, wie oft ein Zeichen in einer Zeichenkette enthalten ist.

### ARGUMENTE

char - Das Zeichen, das gezählt werden soll.

string - Die Zeichenkette, die durchsucht werden soll.

### ERGEBNIS

count - Die Anzahl, wie oft das Zeichen in der Zeichenkette enthalten ist.

### SIEHE AUCH

TR\_FirstOccurance()  
Zeichen in einem Text suchen

## 1.64 TritonRexx.guide/TR\_GetLastError()

Die Funktion TR\_GetLastError() - Zuletzt aufgetretenen Fehler ↔  
ermitteln

\*\*\*\*\*

### NAME

error = TR\_GetLastError(app)

### BESCHREIBUNG

Ermittelt die Nummer des bei einer Applikation zuletzt aufgetretenen Fehlers.

### ARGUMENTE

app - Die Kennzahl der Applikation, deren letzter Fehler ermittelt werden soll.

### ERGEBNIS

error - Die Nummer des zuletzt aufgetretenen Fehlers.

### SIEHE AUCH

TR\_GetErrorString()  
Fehlerbeschreibung anfordern

## 1.65 TritonRexx.guide/TR\_GetErrorString()

Die Funktion TR\_GetErrorString() - Fehlerbeschreibung anfordern

\*\*\*\*\*

### NAME

string = TR\_GetErrorString(error)

### BESCHREIBUNG

Besorgt die Beschreibung eines Fehlers im Klartext.

### ARGUMENTE

error - Die Nummer des Fehlers, diese Nummer kann z.B. mit der Funktion 'TR\_GetLastError()' ermittelt werden.

### ERGEBNIS

string - Die Fehlerbeschreibung im Klartext.

### SIEHE AUCH

TR\_GetLastError()

Zuletzt aufgetretenen Fehler ermitteln

## 1.66 TritonRexx.guide/ASL\_RequestFile()

Die Funktion ASL\_RequestFile() - Datei-Requester aufrufen

\*\*\*\*\*

### NAME

bool = ASL\_RequestFile(project,stemvar,taglist)

### BESCHREIBUNG

Ruft den Datei-Requester der 'asl.library' auf.

### ARGUMENTE

project - Die Kennzahl des Projekts, das für Eingaben gesperrt werden soll während der Requester geöffnet ist. '00000000'x wenn kein Projekt gesperrt werden soll.

stemvar - Stem-Variable, in der die im Requester ausgewählten Datei- oder Verzeichnisnamen gespeichert werden sollen:

stemvar.0 - Anzahl der ausgewählten Dateien  
 stemvar.1 - Pfadname der ersten ausgewählten Datei  
 stemvar.2 - Pfadname der zweiten ausgewählten Datei  
 stemvar.3 - Pfadname der dritten ausgewählten Datei  
 stemvar.n - Pfadname der n'ten ausgewählten Datei

taglist - Eine Tagliste, die die Eigenschaften des Requesters beschreibt:

Kennung	Datenangabe	Beschreibung
ASLFR_PubScreenName	Zeichenkette	Name des öffentlichen Bildschirms, auf dem der Requester geöffnet werden soll.
ASLFR_Screen	Zeiger	Zeiger auf die Screen-Struktur eines Bildschirms, auf dem der Requester geöffnet werden soll.
ASLFR_TextAttr	TextAttr-Struktur	Stem-Variable, die die TextAttr-Struktur für den Requester enthält Siehe
Taglisten verwenden		
.		
ASLFR_TitleText	Zeichenkette	Text in der Titelzeile
ASLFR_PositiveText	Zeichenkette	Text für 'Okay'
ASLFR_NegativeText	Zeichenkette	Text für 'Abbruch'
ASLFR_InitialLeftEdge	Integer	Linke Ecke des Fensters
ASLFR_InitialTopEdge	Integer	Obere Ecke des Fensters
ASLFR_InitialWidth	Integer	Breite des Fensters
ASLFR_InitialHeight	Integer	Höhe des Fensters
ASLFR_InitialFile	Zeichenkette	Dateiname, der beim Öffnen eingetragen werden soll.
ASLFR_InitialDrawer	Zeichenkette	Verzeichnis, das beim Öffnen eingetragen werden soll.
ASLFR_InitialPattern	Zeichenkette	Muster, das beim Öffnen eingetragen werden soll.
ASLFR_DoSaveMode	Bool	1 = Speichern-Modus
ASLFR_DoMultiSelect	Bool	1 = Mehrfachauswahl
ASLFR_DoPatterns	Bool	1 = Muster-Gadget
ASLFR_DrawersOnly	Bool	1 = Verzeichnisauswahl

ASLFR_RejectIcons	Bool	1 = keine Icons anz.
ASLFR_RejectPattern	Zeichenkette	Dateien, auf die das hier angegebene Muster paßt werden nicht angezeigt.
ASLFR_AcceptPattern	Zeichenkette	Nur Dateien, auf die das hier angegebene Muster paßt, werden angezeigt.
ASLFR_FilterDrawers	Bool	1 = auch Verzeichnisse mit den Mustern vergleichen

Zum Erstellen von Taglisten für Standard-Requester stehen einige Funktionen zur Verfügung:

-----

```
LoadFile(title,okay,dir,file)
    Datei zum Laden auswählen

LoadFileI(title,okay,dir,file)
    Datei zum Laden auswählen, Icons anzeigen

LoadFileP(title,okay,dir,file,pattern)
    Datei zum Laden auswählen, Muster-Eingabefeld

LoadFileIP(title,okay,dir,file,pattern)
    Datei zum Laden auswählen, Muster-Eingabefeld, Icons anzeigen

LoadFileM(title,okay,dir,file)
    Mehrere Dateien auswählen

LoadFileMI(title,okay,dir,file)
    Mehrere Dateien auswählen, Icons anzeigen

LoadFileMP(title,okay,dir,file,pattern)
    Mehrere Dateien auswählen, Muster-Eingabefeld

LoadFileMIP(title,okay,dir,file,pattern)
    Mehrere Dateien auswählen, Muster-Eingabefeld, Icons anzeigen

SaveFile(title,okay,dir,file)
    Datei zum Speichern auswählen

SaveFileI(title,okay,dir,file)
    Datei zum Speichern auswählen, Icons anzeigen

SaveFileP(title,okay,dir,file,pattern)
    Datei zum Speichern auswählen, Muster-Eingabefeld

SaveFileIP(title,okay,dir,file,pattern)
    Datei zum Speichern auswählen, Muster-Eingabefeld, Icons ↔
    anzeigen
```

```

GetDrawer(title,okay,dir)
    Verzeichnis auswählen

GetDrawerP(title,okay,dir,pattern)
    Verzeichnis auswählen, Muster-Eingabefeld

```

ERGEBNIS  
 bool - 1 wenn etwas ausgewählt wurde, 0 wenn nicht

SIEHE AUCH  
 -/-

## 1.67 TritonRexx.guide/ASL\_RequestFont()

Die Funktion ASL\_RequestFont() - Font-Requester aufrufen

\*\*\*\*\*

### NAME

```
bool = ASL_RequestFont(project,stemvar,taglist)
```

### BESCHREIBUNG

Ruft den Font-Requester der 'asl.library' auf.

### ARGUMENTE

project - Die Kennzahl des Projekts, das für Eingaben gesperrt werden soll während der Requester geöffnet ist. '00000000'x wenn kein Projekt gesperrt werden soll.

stemvar - Stem-Variable, in der der im Requester ausgewählte Font gespeichert werden soll:

```

stemvar.ta_name - Name des ausgewählten Fonts
stemvar.ta_ysize - Größe des ausgewählten Fonts
stemvar.ta_style - Stil des ausgewählten Fonts
stemvar.ta_flags - Flags des ausgewählten Fonts
stemvar.drawmode - der ausgewählte Zeichenmodus

```

taglist - Eine Tagliste, die die Eigenschaften des Requesters beschreibt:

Kennung	Datenangabe	Beschreibung
ASLFO_PubScreenName	Zeichenkette	Name des öffentlichen Bildschirms, auf dem der Requester geöffnet werden soll.
ASLFO_Screen	Zeiger	Zeiger auf die Screen-Struktur eines Bildschirms, auf dem der Requester geöffnet werden soll.



ASLFO_TextAttr	TextAttr-Struktur	Stem-Variable, die die TextAttr-Struktur für den Requester enthält Siehe Taglisten verwenden .
ASLFO_TitleText	Zeichenkette	Text in der Titelzeile
ASLFO_PositiveText	Zeichenkette	Text für 'Okay'
ASLFO_NegativeText	Zeichenkette	Text für 'Abbruch'
ASLFO_InitialLeftEdge	Integer	Linke Ecke des Fensters
ASLFO_InitialTopEdge	Integer	Obere Ecke des Fensters
ASLFO_InitialWidth	Integer	Breite des Fensters
ASLFO_InitialHeight	Integer	Höhe des Fensters
ASLFO_InitialName	Zeichenkette	Name des Fonts, der beim Öffnen ausgewählt sein soll.
ASLFO_InitialSize	Integer	Größe des Fonts, der beim Öffnen ausgewählt sein soll.
ASLFO_InitialStyle	Integer	Stil des Fonts, der beim Öffnen ausgewählt sein soll.
ASLFO_InitialFlags	Integer	Flags des Fonts, der beim Öffnen ausgewählt sein soll.
ASLFO_InitialDrawMode	Integer	Zeichenmodus des Fonts, der beim Öffnen ausgewählt sein soll.
ASLFO_DoStyle	Bool	1 = Stil abfragen
ASLFO_DoDrawMode	Bool	1 = Zeichenmodus abfragen
ASLFO_FixedWidthOnly	Bool	1 = Nur Fonts anzeigen, die eine feste Zeichenbreite haben.
ASLFO_MinHeight	Integer	Nur Fonts anzeigen, die mindestens die hier angegebene Größe haben
ASLFO_MaxHeight	Integer	Nur Fonts anzeigen, die höchstens die hier

angegebene Größe haben

ASLFO\_ModeList            Einträge            Stem-Variable, die die  
Einträge enthält, die  
im Zeichenmodus-Gadget  
angezeigt werden sollen

Zum Erstellen von Taglisten für Standard-Requester  
stehen einige Funktionen zur Verfügung:

-----

GetFont(title,okay,fontname,size)  
Proportional-Font auswählen

GetFWFont(title,okay,fontname,size)  
Font mit fester Zeichenbreite auswählen

ERGEBNIS

bool        - 1 wenn etwas ausgewählt wurde, 0 wenn nicht

SIEHE AUCH

-/-

## 1.68 TritonRexx.guide/CreateImage()

Die Funktion CreateImage() - Grafik von Disk laden

\*\*\*\*\*

NAME

bool = CreateImage(imagename,filename,taglist)            (Ab Kickstart 3.0)

BESCHREIBUNG

Lädt eine Grafik von Disk und bereitet diese für die Anzeige in einem  
Fenster vor. Jede geladene Grafik muß spätestens am Ende des Skripts  
mit 'DeleteImage()' wieder aus dem Speicher gelöscht werden.

ARGUMENTE

imagename - Ein logischer Name für die Grafik. Der hier angegebene Name  
dient als Parameter für alle Funktionen und Taglisten, bei  
denen die Grafik verwendet werden soll.

Siehe

Taglisten verwenden

.

filename - Der Pfadname der Grafikdatei, die geladen werden soll. Die  
Datei wird mit Hilfe der 'datatypes.library' geladen, es sind  
alle Formate erlaubt, für die Datatypes installiert sind.

taglist - Tagliste, die die Eigenschaften der Grafik beschreibt:

Kennung	Datenangabe	Beschreibung
---------	-------------	--------------

TRXIM_PubScreenName	Zeichenkette	Name eines öffentlichen
---------------------	--------------	-------------------------

		Bildschirms, an dessen Farbpalette die Grafik angepasst werden soll
TRXIM_Screen	Zeiger	Screen-Struktur eines Bildschirms, an dessen Farbpalette die Grafik angepasst werden soll
TRXIM_Flags	Flags	
	TRXIF_CENTER	Die Grafik wird immer im Darstellungsbereich zentriert gezeichnet.  Standard ist die linke, obere Ecke
	TRXIF_SCALE	Die Grafik wird immer genau an die die Größe des Darstellungsbereichs angepasst.  Standardmäßig wird die Größe der Grafik nicht verändert.
	TRXIF_PROP	Beim Anpassen an die Größe des Darstellungsbereichs bleibt das Seitenverhältnis der Grafik immer gleich.  Nur in Verbindung mit dem Flag 'TRIXIF_SCALE'

## ERGEBNIS

bool - 1 wenn alles geklappt hat, 0 im Fehlerfall

## SIEHE AUCH

DeleteImage()  
Grafik aus dem Speicher löschen

ImageWidth()  
Breite einer Grafik ermitteln

ImageHeight()  
Höhe einer Grafik ermitteln

## 1.69 TritonRexx.guide/DeleteImage()

Die Funktion DeleteImage() - Grafik aus dem Speicher löschen

\*\*\*\*\*

NAME

bool = DeleteImage(imagename)

BESCHREIBUNG

Löscht eine zuvor mit 'CreateImage()' geladene Grafik aus dem Speicher. Eine Grafik kann erst gelöscht werden wenn sie von keinem Projekt mehr benutzt wird! 'CreateImage()' kann immer gefahrlos aufgerufen werden, auch wenn keine Grafik mit dem angegebenen Namen vorher geladen wurde!

ARGUMENTE

imagename - Der logische Name der Grafik, die gelöscht werden soll.

ERGEBNIS

bool - 1 wenn die Grafik gelöscht wurde. Wenn die Grafik nicht gelöscht werden konnte wird 0 zurückgegeben. Dies passiert wenn die Grafik noch im Fenster eines Projekts dargestellt wird, oder eine Grafik mit dem angegebenen Namen nicht existiert.

SIEHE AUCH

CreateImage()

Grafik von Disk laden

ImageWidth()

Breite einer Grafik ermitteln

ImageHeight()

Höhe einer Grafik ermitteln

## 1.70 TritonRexx.guide/ImageWidth()

Die Funktion ImageWidth()

\*\*\*\*\*

NAME

width = ImageWidth(imagename)

BESCHREIBUNG

Ermittelt die Breite einer Grafik. Es wird immer die Breite der nicht skalierten Grafik zurückgegeben.

ARGUMENTE

imagename - Der logische Name einer mit 'CreateImage()' geladenen Grafik,

ERGEBNIS

width - Die Breite der Grafik in Pixel

SIEHE AUCH

---

```

CreateImage()
    Grafik von Disk laden

DeleteImage()
    Grafik aus dem Speicher löschen

ImageHeight()
    Höhe einer Grafik ermitteln

```

## 1.71 TritonRexx.guide/ImageHeight()

Die Funktion ImageHeight()

\*\*\*\*\*

### NAME

```
height = ImageHeight(imagename)
```

### BESCHREIBUNG

Ermittelt die Höhe einer Grafik. Es wird immer die Höhe der nicht skalierten Grafik zurückgegeben.

### ARGUMENTE

imagename - Der logische Name einer mit 'CreateImage()' geladenen Grafik

### ERGEBNIS

height - Die Höhe der Grafik in Pixel

### SIEHE AUCH

```

CreateImage()
    Grafik von Disk laden

DeleteImage()
    Grafik aus dem Speicher löschen

ImageWidth()
    Breite einer Grafik ermitteln

```

## 1.72 TritonRexx.guide/TritonRexxVersion()

Die Funktion TritonRexxVersion() - Version der 'tritonrexx.library' ermitteln ↔

\*\*\*\*\*

### NAME

```
version = TritonRexxVersion()
```

#### BESCHREIBUNG

Gibt die Version der 'tritonrexx.library' zurück.

#### ARGUMENTE

-/-

#### ERGEBNIS

version - Die Version der Library, z.B. 37.2

#### SIEHE AUCH

```
TritonVersion()
  Version der 'triton.library' ermitteln
```

## 1.73 TritonRexx.guide/TritonVersion()

Die Funktion TritonVersion() - Version der 'triton.library' ↔  
ermitteln

\*\*\*\*\*

#### NAME

```
version = TritonVersion()
```

#### BESCHREIBUNG

Gibt die Version der 'triton.library' zurück.

#### ARGUMENTE

-/-

#### ERGEBNIS

version - Die Version der Library, z.B. 4.39

#### SIEHE AUCH

```
TritonRexxVersion()
  Version der 'tritonrexx.library' ermitteln
```

## 1.74 TritonRexx.guide/Index

Index und Stichwortverzeichnis

\*\*\*\*\*

Übersicht

---

## Übersicht

ActivateProject()	ActivateProject()
Adresse des Autors	Adresse des Autors
Applikation erzeugen	TR_CreateApp()
Applikation löschen	TR_DeleteApp()
Applikationen	Applikationen und Projekte
Applikationen	Applikationen
Applikationen und Projekte	Applikationen und Projekte
ASL_RequestFile()	ASL_RequestFile()
ASL_RequestFont()	ASL_RequestFont()
Attribute ändern	TR_SetAttribute()
Attribute abfragen	TR_GetAttribute()
Auswahl von Farben	Palette
Autor	Adresse des Autors
Bildschirm freigeben	TR_UnlockScreen()
Bildschirm sperren	TR_LockScreen()
Breite einer Grafik ermitteln	ImageWidth()
Button	Button
CheckBox	CheckBox
Copyright	

---

---

Copyright

CreateImage ()  
CreateImage ()

Cycle  
Cycle

Danksagungen  
Danksagungen

Datei-Requester  
ASL\_RequestFile ()

DeleteImage ()  
DeleteImage ()

DropBox  
DropBox

Eigenschaften von Applikationen  
Applikationen

Eigenschaften von Fenstern  
Projekte-Fenster

Eigenschaften von Menüs  
Menüs

Ein/Aus Schalter  
CheckBox

Einfache Requester  
TR\_EasyRequest ()

Eingaben erlauben  
TR\_UnlockProject ()

Eingaben verbieten  
TR\_LockProject ()

Einrahmen von Gruppen  
FrameBox

Einrahmen von Objekten  
FrameBox

Entstehungsgeschichte  
History

Füllmuster für Fenster  
Füllmuster für Fenster

Füllmuster für Objekte  
Füllmuster für Objekte

Farbauswahl

---



## Palette

Fehler ermitteln  
TR\_GetLastError()

Fehlerbeschreibung ermitteln  
TR\_GetErrorString()

Felder zur Mehrfachauswahl  
Cycle

Fenster öffnen  
TR\_OpenProject()

Fenster übernehmen  
TR\_ObtainWindow()

Fenster als Requester  
TR\_AutoRequest()

Fenster eines Projekts aktivieren  
ActivateProject()

Fenster eines Projekts nach hinten  
ProjectToBack()

Fenster eines Projekts nach vorne  
ProjectToFront()

Fenster freigeben  
TR\_UnlockProject()

Fenster freigeben  
TR\_ReleaseWindow()

Fenster schließen  
TR\_CloseProject()

Fenster sperren  
TR\_LockProject()

Fenster-Füllmuster  
Füllmuster für Fenster

Flächen zum Fallenlassen von Icons  
DropBox

FlushTRX  
FlushTRX

Font-Requester  
ASL\_RequestFont()

Fortschrittsanzeige  
Progress

FrameBox

---

## FrameBox

```
Funktion ActivateProject ()
    ActivateProject ()

Funktion ASL_RequestFile ()
    ASL_RequestFile ()

Funktion ASL_RequestFont ()
    ASL_RequestFont ()

Funktion CreateImage ()
    CreateImage ()

Funktion DeleteImage ()
    DeleteImage ()

Funktion ImageHeight ()
    ImageHeight ()

Funktion ImageWidth ()
    ImageWidth ()

Funktion ProjectToBack ()
    ProjectToBack ()

Funktion ProjectToFront ()
    ProjectToFront ()

Funktion TritonRexxVersion ()
    TritonRexxVersion ()

Funktion TritonVersion ()
    TritonVersion ()

Funktion TR_AutoRequest ()
    TR_AutoRequest ()

Funktion TR_CloseProject ()
    TR_CloseProject ()

Funktion TR_CreateApp ()
    TR_CreateApp ()

Funktion TR_DeleteApp ()
    TR_DeleteApp ()

Funktion TR_EasyRequest ()
    TR_EasyRequest ()

Funktion TR_FirstOccurance ()
    TR_FirstOccurance ()

Funktion TR_GetAttribute ()
    TR_GetAttribute ()

Funktion TR_GetErrorString ()
```

---

```
TR_GetErrorString()

Funktion TR_GetLastError()
    TR_GetLastError()

Funktion TR_GetMsg()
    TR_GetMsg()

Funktion TR_HandleMsg()
    TR_HandleMsg()

Funktion TR_LockProject()
    TR_LockProject()

Funktion TR_LockScreen()
    TR_LockScreen()

Funktion TR_NumOccurrences()
    TR_NumOccurrences()

Funktion TR_ObtainWindow()
    TR_ObtainWindow()

Funktion TR_OpenProject()
    TR_OpenProject()

Funktion TR_ReleaseWindow()
    TR_ReleaseWindow()

Funktion TR_ReplyMsg()
    TR_ReplyMsg()

Funktion TR_SendMessage()
    TR_SendMessage()

Funktion TR_SetAttribute()
    TR_SetAttribute()

Funktion TR_UnlockProject()
    TR_UnlockProject()

Funktion TR_UnlockScreen()
    TR_UnlockScreen()

Funktion TR_Wait()
    TR_Wait()
```

Funktionen

Funktionen

Funktionen der Library  
Funktionen

Gemeinsamkeiten mit der 'triton.library'  
triton.library

Grafik löschen

---

---

	DeleteImage ()
Grafik von Disk laden	CreateImage ()
Grafiken	Image
Group	Group
Gruppen	Group
Gruppen einrahmen	FrameBox
Höhe einer Grafik ermitteln	ImageHeight ()
History	History
Image	Image
Image löschen	DeleteImage ()
Image von Disk laden	CreateImage ()
ImageHeight ()	ImageHeight ()
ImageWidth ()	ImageWidth ()
Index	Index
Informationen über 'Triton'	Triton-Info
Installation	Installation
Knöpfe	Button
Kurze Vorstellung der Library	Übersicht
Kurzhilfe	QuickHelp
Leerräume zwischen Objekten	

---

---

Space	
Line	Line
Listen zur Darstellung und Auswahl Listview	
Listview	Listview
Mehrfachauswahl	Cycle
Menüs	Menüs
Nachricht beantworten TR_Wait()	
Nachricht beantworten TR_ReplyMsg()	
Nachricht empfangen TR_GetMsg()	
Nachricht empfangen und beantworten TR_HandleMsg()	
Nachrichten an Objekte TR_SendMessage()	
Oberflächen erstellen Oberflächen erstellen	
Objekt-Füllmuster Füllmuster für Objekte	
Objekte einrahmen FrameBox	
Objekte und Attribute Objekte und Attribute	
Palette	Palette
Progress	Progress
ProjectToBack()	ProjectToBack()
ProjectToFront()	ProjectToFront()
Projekt öffnen	

---

---

	TR_OpenProject ()
Projekt freigeben	TR_UnlockProject ()
Projekt schließen	TR_CloseProject ()
Projekt sperren	TR_LockProject ()
Projekte	Applikationen und Projekte
Projekte/Fenster	Projekte-Fenster
QuickHelp	QuickHelp
Rechliches	Copyright
Schieberegler zum Scrollen durch Listen	
Scroller	
Schieberegler zur Einstellung von Werten	
Slider	
Screen-Zeiger ermitteln	TR_LockScreen ()
Scroller	Scroller
Slider	Slider
Space	Space
Stichwortverzeichnis	
Index	
String	String
Support-Mailbox	Adresse des Autors
Systemvoraussetzungen	Systemvoraussetzungen
Taglisten verwenden	Taglisten verwenden
Text	

---

## Text

Texte in verschiedenen Darstellungsarten  
Text

Texteingabefelder  
String

Tips und Problemlösungen  
Tips und Problemlösungen

Trennlinien  
Line

Triton-Info  
Triton-Info

Triton-Version ermitteln  
TritonVersion()

triton.library  
triton.library

TritonRexx-Version ermitteln  
TritonRexxVersion()

TritonRexxVersion()  
TritonRexxVersion()

TritonVersion()  
TritonVersion()

TR\_AutoRequest()  
TR\_AutoRequest()

TR\_CloseProject()  
TR\_CloseProject()

TR\_CreateApp()  
TR\_CreateApp()

TR\_DeleteApp()  
TR\_DeleteApp()

TR\_EasyRequest()  
TR\_EasyRequest()

TR\_FirstOccurance()  
TR\_FirstOccurance()

TR\_GetAttribute()  
TR\_GetAttribute()

TR\_GetErrorString()  
TR\_GetErrorString()

TR\_GetLastError()

---

---

	TR_GetLastError()
TR_GetMsg()	TR_GetMsg()
TR_HandleMsg()	TR_HandleMsg()
TR_LockProject()	TR_LockProject()
TR_LockScreen()	TR_LockScreen()
TR_NumOccurrences()	TR_NumOccurrences()
TR_ObtainWindow()	TR_ObtainWindow()
TR_OpenProject()	TR_OpenProject()
TR_ReleaseWindow()	TR_ReleaseWindow()
TR_ReplyMsg()	TR_ReplyMsg()
TR_SendMessage()	TR_SendMessage()
TR_SetAttribute()	TR_SetAttribute()
TR_UnlockProject()	TR_UnlockProject()
TR_UnlockScreen()	TR_UnlockScreen()
TR_Wait()	TR_Wait()
Was tun wenn...	Tips und Problemlösungen
Window-Zeiger ermitteln	TR_ObtainWindow()
Zeichen suchen	TR_FirstOccurance()
Zeichen zählen	TR_NumOccurrences()

---