

in

Georg Steger

COLLABORATORS

	<i>TITLE :</i> in		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Georg Steger	January 19, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	in	1
1.1	ReqAttack Tools	1
1.2	Einleitung	1
1.3	Features	1
1.4	Installation	2
1.5	Benutzung	3
1.6	Konfiguration	4
1.7	Konfiguration Variablen	7
1.8	Konfiguration - Allgemeine Einstellungen	10
1.9	Konfiguration - Requester Look	11
1.10	Konfiguration - Button Look	13
1.11	Konfiguration - Requester Handling	18
1.12	Konfiguration - Requester Layout	20
1.13	Konfiguration - Check Variablen	23
1.14	DataTypes	25
1.15	Bugs	26
1.16	Trouble Shooting	26
1.17	Geschichte	26
1.18	Zukunft	28
1.19	Autor	29
1.20	Danksagungen	29
1.21	Gesetzliches	29
1.22	Abschließend	30

Chapter 1

in

1.1 ReqAttack Tools

ReqAttack 1.0 - (C) 1999 by Georg Steger

Einleitung	Trouble Shooting
Features	Geschichte
Installation	Zukunft
Benutzung	
Konfiguration	Autor
Tools	Danksagungen
DataTypes	Gesetzliches
Bugs	Abschließend

1.2 Einleitung

ReqAttack ist ein kleines Programm, welches das Aussehen und die Bedienung von Requestern (Dialogfenstern) all jener Programmen verbessert, die die standard AmigaOS Funktionen (EasyRequesterArgs, AutoRequest, BuildEasyRequestArgs, BuildSysRequest) dazu verwenden.

***** ReqAttack ist Shareware!! Lesen Sie Gesetzliches ! *****

1.3 Features

- o Konfigurierbare Requester Logos, die auch animiert sein können
 - o Konfigurierbare Button Images.
 - o Konfigurierbares Button-Aussehen.
 - o Konfigurierbares Requester-Hintergrundmuster.
 - o Tastaturbedienung. Man kann mit TAB, SHIFT TAB und den Cursor Tasten durch die Buttons cyclen. Der hervorgehobene Button kann mit RETURN und einer weiteren Taste, die konfigurierbar ist (Vorgabe LCOMMAND v), aktiviert
-

werden. Der Button ganz rechts kann mit ESCAPE und einer weiteren, konfigurierbaren (Vorgabe LCOMMAND b) Taste aktiviert wrden. Es ist möglich, die tastaturmäßige Aktivierung von Buttons abzubrechen, indem man eine der Umschalttasten oder ESCAPE drückt, bevor die Button-Aktivierungs-Taste losgelassen wird.

- o Möglichkeit, die Requester zu verschieben, indem man in eine beliebige Stelle des Fensters klickt.
- o Möglichkeit, die Requester in der Mitte des Bildschirms oder unter dem Mauszeiger erscheinen zu lassen.
- o Sehr spezielle bedingte Konfiguration.
- o Man kann fast alles konfigurieren, wie zum Beispiel den Abstand zwischen den einzelnen Elementen des Requester, deren Ausrichtung, die Farbe des Hervorhebungs Rechtecks und vieles mehr.
- o Commodities Schnittstelle.

1.4 Installation

Klicken Sie doppelt auf das Install Icon, um ReqAttack auf Ihrer Festplatte zu installieren. Wenn bei der Installation Probleme auftreten (Abbruch der Installation von seiten des Installers mit irgendeiner Fehlermeldung), überprüfen Sie bitte die Version des Installers und wenn notwendig holen Sie sich die neueste Version (>= 43.3) des Installers vom Aminet.

Das Installationsskript wird ReqAttack nicht in Ihre Startup-Sequence, User-Startup oder WBStartup einbinden. Ziehen Sie das ReqAttack Icon in die WB-Startup Schublade, wenn ReqAttack während des Bootens gestartet werden soll, aber erst nach erfolgter Installation! Wenn Sie bereits eine ältere (evtl. nicht öffentliche Beta) Version von ReqAttack benutzten, dann sorgt das Installationsskript automatisch dafür, daß ein sich eventuell in der WBStartup Schublade befindendes ReqAttack Executabl e mit der neuen Version ersetzt wird.

Wenn Sie die ReqAttack Schublade nach der Installation verschieben wollen, ändern Sie die Datei "S:ReqAttack.installinfo" mit einem Text Editor, um dort den Pfad anzupassen. Wenn Sie das nicht machen, kann es vorkommen, daß zukünftige Updates von ReqAttack bei der Installation Probleme bereiten!

Wenn Sie ReqAttack de-installieren wollen, dann müssen Sie das leider von Hand machen, weil es zur Zeit kein De-Installationsskript gibt:

Hauptprogramm, Tools, Dokumentation und Images:

- o Löschen Sie die ReqAttack Schublade
- o Löschen Sie das ReqAttack Programm in der WBStartup Schublade, falls Sie es dorthin gezogen/kopiert haben

ReqAttack Konfigurationsdatei:

- o Löschen Sie "ENVARC:ReqAttack.prefs"

ReqAttack Installations-Info-Datei:

- o Löschen Sie "S:ReqAttack.installinfo"

ReqAttack DataTypes:

- o Löschen Sie "SYS:Classes/DataTypes/rabrush.datatype"
Löschen Sie "SYS:Classes/DataTypes/raanimbrush.datatype"
- o Löschen Sie "SYS:Devs/DataTypes/RABrush"
Löschen Sie "SYS:Devs/DataTypes/RABrush.info"
Löschen Sie "SYS:Devs/DataTypes/RAAnimBrush"
Löschen Sie "SYS:Devs/DataTypes/RAAnimBrush.info"

CyReq (wird evtl. von anderen Programmen, wie PowerWindows benötigt!!):

- o Löschen Sie "SYS:C/CyReq"
- o Entfernen Sie die "CyReq" Zeile aus der "S:Startup-Sequence"

1.5 Benutzung

ReqAttack kann sowohl von einer Shell als auch von der Workbench aus gestartet werden. Die folgenden Shell Argumente bzw. Workbench Tooltypes können angegeben werden:

HOMEDIR: Damit kann man festlegen, welches aktuelle Verzeichnis ReqAttack verwenden soll, während das Programm läuft. Dadurch ist es möglich relative Pfade bei Image Dateien auch dann zu verwenden, wenn man ReqAttack in die WBStartup Schublade geschoben hat, wo das normale aktuelle Verzeichnis für ReqAttack ja "SYS:WBStartup" sein würde.

Das Installationsskript setzt automatisch das HOMEDIR Workbench Tooltype auf das Ziel Verzeichnis, das Sie für ReqAttack ausgewählt haben.

CYREQ : ReqAttack kann bei älteren Version von CyReq nicht automatisch erkennen, ob es aktiviert ist. Die neue Version, welche zusammen mit diesem Archiv geliefert wird, und die Sie automatisch durch das Installationsskript installieren lassen können, ermöglicht Autodetektion. Wenn Sie aus irgendeinem Grund weiterhin eine ältere Version von CyReq benutzen wollen, dann müssen Sie diese Option verwenden, andernfalls wird CyReq keinerlei Effekt haben.

Diese Option macht natürlich nur dann Sinn, wenn Sie CyReq auch wirklich in der Startup-Sequence verwenden. Für weitere Informationen lesen Sie die CyReq Dokumentation .

Um ReqAttack zu beenden, schicken Sie dem Programm ein Break Signal (CONTROL C) oder starten Sie das Programm ein zweites Mal. Eine andere Möglichkeit besteht darin, Exchange zu verwenden. Seit Version 1.0 besitzt ReqAttack eine Commodities Schnittstelle.

Wenn das Beenden nicht funktioniert, werden Sie nur einen Flash auf dem

Bildschirm sehen und ReqAttack wird weiterlaufen. Das kann passieren, wenn andere Programme OS Funktionen gepatcht haben, die ReqAttack bereits vor diesen Programmen gepatcht hat. In so einem Fall sollte es möglich sein, ReqAttack zu beenden, indem man zuerst diese anderen Programme beendet.

Es ist auch möglich, ReqAttack vorübergehend zu deaktivieren, indem man in Exchange das Cycle Gadget von "Aktiv" auf "Inaktiv" stellt, nachdem man ReqAttack in der Liste links angewählt hat. Danach werden neu erzeugte Requester nicht mehr von ReqAttack verändert, solange bis Sie das Cycle Gadget wieder auf "Aktiv" stellen.

1.6 Konfiguration

Quick Links: Konfigurations Variablen Check Variablen

Leider gibt es kein Einstellungsprogramm und es ist unsicher, ob es jemals eines geben wird, weil die Konfiguration irgendwie besonders ist. Ein anderer Grund ist, daß ich es wirklich hasse, Einstellungsprogramme zu schreiben ...

Es gibt zwei Konfigurations Dateien, die beide "ReqAttack.prefs" heißen. Eine davon befindet sich im ENV: Verzeichnis, welches üblicherweise in der RAM Disk ist. Die andere befindet sich im ENVARC: Verzeichnis, welches normalerweise auf dem Boot Volumen Ihrer Festplatte ist. Die Tatsache, daß es zwei Konfigurations Dateien gibt, ist nichts besonderes. Viele andere Programme wie die standard Workbench Einstellprogramme funktionieren auf die gleiche Weise. Der Unterschied ist, daß diese Programme eine Oberfläche (GUI) haben, und man es deshalb nicht mitbekommt.

Die "ReqAttack.prefs" Datei in ENVARC: ist die wichtigere von beiden. Das Editieren dieser Datei entspricht dem Klicken auf "Speichern" in einem der Workbench Einstellprogramme. Während des Bootens wird diese Datei von der Startup-Sequence nach ENV: kopiert. Deshalb befindet sich dort anfangs, nach dem Booten, stets eine exakte Kopie der gespeicherten Konfigurations Datei.

Das Editieren von "ReqAttack.prefs" in ENV: entspricht dem Klicken auf "Benutzen" in einem der Workbench Einstellprogramme. ReqAttack liest die Konfigurationsdatei aus ENV: beim Start und jedesmal, wenn diese Datei in ENV: geändert wird, d. h. wenn Sie sie von Ihrem Editor aus speichern.

Um ReqAttack zu konfigurieren, werden Sie also hauptsächlich die Datei ENV:ReqAttack.prefs editieren. Wenn Ihnen die aktuelle Konfiguration dann gefällt, und Sie deshalb permanent speichern wollen – die Dateien in ENV: sind nach einem Reset verloren – kopieren Sie einfach ENV:ReqAttack.prefs nach ENVARC:

Die Konfigurations Datei ist eine ganz einfache Textdatei. Zeilen, die mit einem ';' Zeichen beginnen werden als Kommentare betrachtet und deshalb ignoriert. Vor dem ';' Zeichen dürfen sich auch Leerzeichen und Tabs befinden.

Das besondere an ReqAttack's Konfigurationsdatei ist, daß sie wie ein Skript ist, das jedesmal ausgeführt wird, wenn ein Requester geöffnet wird. Man kann Bedingungen einbauen, und dadurch sieht das ganze fast ein bißchen wie

eine Programmiersprache aus, mit einer ähnlichen Syntax wie BASIC. Bevor wir uns den Bedingungen zuwenden, laßt uns sehen, wie eine normale Zeile aussieht, wo man einer bestimmten Konfigurations Variable einen Wert zuweist:

```
KFGVARIABLE = WERT
```

Vor und nach dem '=' Zeichen muß mindestens ein Leerzeichen oder Tab sein! WERT ist immer konstant und kann je nach KFGVARIABLE entweder ein String (Text), eine Zahl oder ein Spezialname sein. Zahlen (numerische Werte) kann man auch in hexadezimaler Schreibweise angeben, indem man "0x" oder "0X" verwendet, z. B. 0x100. Strings sollten in Anführungszeichen eingeschlossen sein. Falls der String Leerzeichen enthält ist das unbedingt notwendig. Die Groß-/ Kleinschreibung von KFGVARIABLE ist egal, man kann sogar Unterstriche ('_') einfügen, um lange KFGVARIABLEn lesbarer zu machen.

Liste aller Konfigurations Variablen

Jetzt sehen wir uns an, wie man Bedingungen in die Konfigurations Datei einbaut. Das ist sehr einfach, wie man im folgenden Beispiel sieht:

```
if taskname = "Workbench"
    logo = "cool_wb_logo"
endif
```

"If" ist das englische Wort für "wenn", und "endif" steht für "Ende wenn" oder man könnte auch sagen "Ende Bedingung". In diesem Beispiel wird der Konfigurations Variablen "LOGO" ein Wert zugewiesen, nämlich "cool_wb_logo", aber nur dann, wenn der Requester von der Workbench geöffnet wird. Die "logo =" Zeile ist eingerückt, damit alles lesbarer wird. Jetzt wollen wir die Konfigurations Datei so erweitern, daß alle Requester, die *nicht* von der Workbench geöffnet wurden, das Logo "standard_logo" verwenden. Wir könnten die Konfigurationsdatei wie folgt schreiben:

```
logo = "standard_logo"
if taskname = "Workbench"
    logo = "cool_wb_logo"
endif
```

Das funktioniert, besser ist aber:

```
if taskname = "Workbench"
    logo = "cool_wb_logo"
else
    logo = "standard_logo"
endif
```

"Else" ist das englische Wort für "sonst", "wenn nicht", "andernfalls". Natürlich kann man mehrere Zeilen zwischen IF und ENDIF (und zwischen IF und ELSE - und zwischen ELSE und ENDIF) haben. Auch Verschachtelungen sind kein Problem:

```
if taskname = "Workbench"
    logo = "cool_wb_logo"
    hilightpen = 0xFFFF00
else
    logo = "standard_logo"
    if screendepth < 2
```



```

        highlightpen = SHADOWPEN
    else
        highlightpen = 0xFF0000
    endif
    reqpos = MOUSE
endif

```

Daraus folgt, daß jede Bedingung wie folgt geschrieben werden muß:

```

IF CHECKVARIABLE VERGLEICHszeichen WERT
    ;Zeilen, die ausgeführt werden, wenn die Bedingung zutrifft.
ELSE
    ;Zeilen, die ausgeführt werden, wenn die Bedingung nicht zutrifft.
ENDIF

```

ELSE und die Zeilen zwischen ihm und ENDIF sind optional, man kann sie also auch weglassen. Mögliche VERGLEICHszeichen sind:

```

= Bedingung trifft zu, wenn CHECKVARIABLE gleich WERT ist. Anstelle von
  "=" kann man auch "==" schreiben, wie in C.

<> Bedingung trifft zu, wenn CHECKVARIABLE ungleich WERT ist. Anstelle
  von "<>" kann man auch "!=" schreiben, wie in C.

< Bedingung trifft zu, wenn CHECKVARIABLE kleiner WERT ist.

> Bedingung trifft zu, wenn CHECKVARIABLE größer WERT ist.

<= Bedingung trifft zu, wenn CHECKVARIABLE kleiner gleich WERT ist.

>= Bedingung trifft zu, wenn CHECKVARIABLE größer gleich WERT ist.

```

IN Bedingung trifft zu, wenn CHECKVARIABLE gleich einem der durch Kommas getrennten Unter-WERTE ist. Bei IN muß WERT auf jeden Fall in Anführungszeichen eingeschlossen sein. Zum Beispiel:

```

IF screendepth IN "2,4,6,8"
    logo = "irgendwas"
ENDIF

```

Diese Bedingung trifft zu, wenn die Bildschirm Tiefe entweder 2, 4, 6 oder 8 Bit ist. Bei CHECK Variablen, die eine Zeichenkette zurückgeben, kann jeder Unter-WERT natürlich auch ein AmigaDOS Muster sein:

```

IF taskname IN "#?multiview#?,#?Workbench#?"
    logo = "irgendwas"
ENDIF

```

Liste aller Check Variablen

Seit Version 1.1 ist es möglich, das Ausführen der Konfigurationsdatei an jeder beliebigen Stelle durch den Befehl STOP zu beenden. Das kann in manchen Situationen übersichtlicher und kürzer sein, als tief verschachtelte Bedingungen zu verwenden. So kann man anstelle von ...

```

IF taskname = "Workbench"
    logo = "wblogo"

```

```

ELSE
    buttonstyle = KDE
    IF taskname = "More"
        logo = "morelogo"
    ELSE
        nobuttonimages = TRUE
        IF taskname = "Calculator"
            logo = "calclogo"
        ENDIF
    ENDIF
ENDIF

```

... das ganze auch so schreiben:

```

IF taskname = "Workbench"
    logo = "wblogo"
    STOP
ENDIF

buttonstyle = KDE

IF taskname = "More"
    logo = "morelogo"
    STOP
ENDIF

nobuttonimages = TRUE

IF taskname = "Calculator"
    logo = "calclogo"
    STOP
ENDIF

```

Abschließend noch ein paar wichtige Anmerkungen zu Fehlern (Syntax Fehler, IF ohne ENDIF, usw.) in der Konfigurations Datei. Wenn ReqAttack gestartet wird, und es gibt einen Fehler in der Konfigurationsdatei, dann wird das Programm beendet und eine Fehlermeldung angezeigt. Wenn ReqAttack beim Start keine Fehler meldet und Sie später "ENV:ReqAttack.prefs" editieren, dann verursachen Fehler in der Konfigurationsdatei nicht das automatische Beenden des Programms. Stattdessen werden Sie nur einen Screenflash sehen, und ReqAttack wird auf die vorherige (funktionierende) Konfiguration zurückschalten!

1.7 Konfiguration Variablen

Es folgt eine Liste aller Konfigurations Variablen . Klicken Sie auf einen der Links, um nähere Informationen zu der entsprechenden Variable zu erhalten, z. B. mögliche Werte usw. Welche Art von Wert eine bestimmte Variable erwartet wird zwischen Klammern angezeigt.

BOOL Variablen können entweder positiv ("wahr") oder negativ ("falsch") sein. Dementsprechend kann man die englischen Wörter "YES", "TRUE", "ON" (positiv) bzw. "NO", "FALSE", "OFF" (negativ) verwenden.

Den Variablen, die einen Image Dateinamen erwarten, kann auch eines der ins

Programm eingebauten standard Images zugewiesen werden, indem man "\$INFO-LOGO", "\$ASKLOGO", "\$MULTIASKLOGO", "\$OKIMAGE", "\$CANCELIMAGE" oder "\$THIRD-IMAGE" als Wert angibt. Bitte beachten Sie, daß bei externen Images nur das ReqAttack eigene Dateiformat unterstützt wird. Benutzen Sie das Shell Tool RAIM_Convert für Konvertierungszwecke und lesen Sie dessen Dokumentation.

Bei Variablen, die eine Farbe erwarten, kann man entweder einen 24 Bit RGB Wert (indem man den numerischen Wert im dezimalen oder hexadezimalen Format schreibt), einen direkte Colormap Stift (indem man ein '#' Zeichen vor der Stiftnummer verwendet, z. B. würde #0 die erste Farbe aus der Palette des Screens bedeuten) oder einen der folgenden Spezialnamen angeben. Diese Spezialnamen veranlassen ReqAttack dazu, den entsprechenden DrawInfo Stift vom Screen, auf dem der Requester geöffnet wird, zu verwenden:

```
? DETAILPEN
? BLOCKPEN
  TEXTPEN
  SHINEPEN
  SHADOWPEN
  FILLPEN
  FILLTEXTPEN
? BACKGROUNDPEN
  HIGHLIGHTTEXTPEN
? BARDETAILPEN
? BARBLOCKPEN
? BARTRIMPEN
```

? AmigaGuide DataType kann diese Farben nicht darstellen

Wenn man die Farbe als RGB Wert angibt, sollte man, weil es am einfachsten ist, die hexadezimale Schreibweise verwenden. Für eine gelbe Farbe könnte man zum Beispiel "0xFFFF00" angeben.

ALLGEMEINE EINSTELLUNGEN

IMAGECACHES	Anzahl der im Speicher zu cachenden Images
BITMAPFLAGS	Besonders interessant für Grafikkarten Besitzer
DONOTPATCH	Requester soll nicht von ReqAttack gepatcht werden

REQUESTER LOOK

INFOLOGO	Vorgabelogo für Requester mit einem Button
ASKLOGO	Vorgabelogo für Requester mit zwei Buttons
MULTIASKLOGO	Vorgabelogo für Requester mit zwei oder mehr Buttons
LOGO	Logo, das verwendet werden soll
NOLOGO	Kein Logo verwenden
BACKGROUND	Image für Hintergrundmuster
BGPATTERN1PEN	Hintergrundmuster Farbe 1
BGPATTERN2PEN	Hintergrundmuster Farbe 2
TEXTBOXBGPEN	Hintergrundfarbe der Box mit dem Requestertext
TEXTBOXTEXTPEN	Farbe des Requestertextes
OTEXTBOXEDGE1PEN	Farbe der äußeren Textboxkanten links und oben
OTEXTBOXEDGE2PEN	Farbe der äußeren Textboxkanten rechts und unten

ITEXTBOXEDGE1PEN	Farbe der inneren Textboxkanten links und oben
ITEXTBOXEDGE2PEN	Farbe der inneren Textboxkanten rechts und unten

BUTTON LOOK

BUTTONSTYLE	Vordefinierte Button Stile
OBUTTONEDGE1PEN	Farbe der äußeren Buttonkanten links und oben
OBUTTONEDGE2PEN	Farbe der äußeren Buttonkanten rechts und unten
IBUTTONEDGE1PEN	Farbe der inneren Buttonkanten links und oben
IBUTTONEDGE2PEN	Farbe der inneren Buttonkanten rechts und unten
OBUTTONEDGE1SELPEN	Farbe der äußeren Buttonkanten links und oben (sel.)
OBUTTONEDGE2SELPEN	Farbe der äußeren Buttonkanten rechts und unten (sel.)
IBUTTONEDGE1SELPEN	Farbe der inneren Buttonkanten links und oben (sel.)
IBUTTONEDGE2SELPEN	Farbe der inneren Buttonkanten rechts und unten (sel.)
OBUTTONEDGE1HIPEN	Farbe der äußeren Buttonkanten links und oben (hvg.)
OBUTTONEDGE2HIPEN	Farbe der äußeren Buttonkanten rechts und unten (hvg.)
IBUTTONEDGE1HIPEN	Farbe der inneren Buttonkanten links und oben (hvg.)
IBUTTONEDGE2HIPEN	Farbe der inneren Buttonkanten rechts und unten (hvg.)
BUTTONBGPEN	Hintergrundfarbe der Buttons
BUTTONBGSELPEN	Hintergrundfarbe selektierter Buttons
BUTTONBGHIPEN	Hintergrundfarbe hervorgehobener Buttons
BUTTONTEXTPEN	Textfarbe der Buttons
BUTTONTEXTSELPEN	Textfarbe selektierter Buttons
BUTTONTEXTHIPEN	Textfarbe hervorgehobener Buttons
BUTTONTEXTSTYLE	Textstil der Buttons
BUTTONTEXTSELSTYLE	Textstil selektierter Buttons
BUTTONTEXTHISTYLE	Textstil hervorgehobener Buttons
SHIFTBUTTONTEXT	Button Text (und Image) 1 Pixel scrollen wenn sel.
BUTTONISPACINGX	Horiz. Abstand zwischen Buttonrahmen und Text/Image
BUTTONISPACINGY	Min. vert. Abstand zwischen Buttonrahmen und Text
BUTTONISPACINGY2	Min. vert. Abstand zwischen Buttonrahmen und Image
BUTTONEXTRAWIDTH	Wird zur Mindestbreite der Buttons dazugezählt
BIMAGETEXTSPACING	Min. Abstand zwischen Button Image und Text
HILIGHTPEN	Farbe des Hervorhebungs-Rechtecks
HITHICKX	Horizontale Dicke des Hervorhebungs-Rechtecks
HITHICKY	Vertikale Dicke des Hervorhebungs-Rechtecks
OKIMAGE	Image für den Button ganz links (OK)
CANCELIMAGE	Image für den Button ganz rechts (Abbrechen)
THIRDIMAGE	Image für alle anderen Buttons
NOBUTTONIMAGES	Keine Button Images verwenden

REQUESTER HANDLING

OKKEY	Tastaturkürzel für Button ganz links (OK)
CANCELKEY	Tastaturkürzel für Button ganz rechts (Abbrechen)
KEYQUALIFIER	Was muß für Tastaturkürzel zusätzlich gedrückt sein
FUNCTIONKEYS	Möglichkeit, Buttons mit Funktionstasten zu aktivieren
COOLDRAGGING	Verschieben der Req. durch Klick an beliebiger Stelle

REQUESTER LAYOUT

OUTERSPACINGX	Horiz. Abst. zwischen Fensterrand und Requesterinhalt
OUTERSPACINGY	Vert. Abst. zwischen Fensterrand und Requesterinhalt
INNERSPACINGX	Min. horiz. Abst. zwischen Requesterboxrahmen und Text

INNERSPACINGY	Min. vert. Abst. zwischen Requesterboxrahmen und Text
BODYGADGETSPACING	Abstand zwischen Requesterbox und den Buttons
BUTTONSPACING	Minimaler Abstand zwischen den Buttons
BODYLOGOSPCING	Minimaler Abstand zwischen Logo und Requestertext
LOGOALIGN	Vertikale Ausrichtung des Logos
BODYXALIGN	Horizontale Ausrichtung des Requestertextes
BODYYALIGN	Vertikale Ausrichtung des Requestertextes
REQTEXTALIGN	Ausrichtung der Requestertext Zeilen
BUTTONALIGN	Horizontale Ausrichtung der Buttons
SAMEBUTTONWIDTH	Gleiche Breite für alle Buttons
REQPOS	Position, an der der Requester erscheint

1.8 Konfiguration - Allgemeine Einstellungen

```
+-----+
| IMAGECACHES (NUMERISCH)                                     (Vorgabe = 5) |
+-----+
```

Anzahl der (externen) Images, die im Speicher zwischengespeichert werden sollen, damit sie nicht jedesmal neu geladen werden müssen. ReqAttack cached nicht Bitmaps, sondern Dateien!

```
+-----+
| BITMAPFLAGS (..)                                           (Vorgabe = MF) |
+-----+
```

Hiermit kann festgelegt werden, in welchem Format ReqAttack die internen Button Image und Requester Logo Bitmaps erzeugen soll, was besonders für Grafikkarten Besitzer interessant ist. Mögliche WERTE sind eine Kombination von:

I: BMF_INTERLEAVED (wenn möglich immer interleavte Bitmaps erzeugen)
M: BMF_MINPLANES (gleiche Farbtiefe wie Screen verwenden)
F: FRIEND (Bitmap soll gleiches Format wie Screen Bitmap haben)

Normalerweise wird beim Erzeugen von Bitmaps für Images, die eine Maske besitzen (Button Images), das Erzeugen einer interleavten Bitmap verhindert, wenn der Screen ein standard Amiga Screen ist. Das wird deshalb gemacht, weil bei interleavten Bitmaps die Maske viel mehr Speicher benötigt. Interleavte Bitmaps haben zwar den Vorteil, daß beim Blitten weniger/kaum Blinkeffekte zu sehen sind, aber da die Button Images recht klein und nicht animiert sind, merkt man das kaum. Deshalb ist 'I' vorgabemäßig nicht eingeschaltet. Für Grafikkarten Screens ist 'I' zudem ohne Bedeutung.

'M' macht nur in Verbindung mit 'F' Sinn. 'F' erzwingt eine Bitmap im Chunky Format (falls der Screen ein Grafikkarten Screen ist). Wird diese Option nicht verwendet, so wird immer eine planare Bitmap im CHIP RAM erzeugt, was zu Blinkeffekten führen kann. Wird die Option 'M' zusätzlich zu 'F' angegeben, dann wird zusätzlich die Farbtiefe des Screens übernommen, womit auf 15/16 (HiColor) und 24 (TrueColor) Bit Screens das Farb-Remapping entfällt, was zu einer höheren Qualität, aber auch mehr Speicherverbrauch führt.

ReqAttack verwendete in Versionen <= 1.0 keine dieser Flags (BITMAPFLAGS = ""), d. h. es wurden immer planare nicht interleavte Bitmaps erzeugt.

```
+-----+
| DONOTPATCH (BOOL)                                     (Vorgabe = NO) |
+-----+
```

Bei bestimmten Requestern kann es eventuell zu Problemen kommen, wenn ReqAttack läuft. Wenn z. B. aus irgendeinem Grund ein OS Funktionsaufruf von ReqAttack das Erscheinen eines Requesters hervorruft (z. B. durch Dateisystem Prozesse ausgelöst, die pr_WindowPtr nicht beachten) dann kann eine Deadlock Situation entstehen, die sowohl ReqAttack als auch den Task, der den Requester erzeugen will, lahm legen. Kein Task wird mehr Requester anzeigen können (wenn er das versucht wird er ebenfalls blockiert), und beenden kann man in so einem Fall ReqAttack auch nicht mehr. Es bleibt einem also nichts anderes übrig, als den Computer neu zu booten. Um solche Probleme zu vermeiden, kann man ReqAttack dazu veranlassen, bestimmte Requester nicht zu patchen, d. h. die original Requester Routinen aus dem OS wieder zum Zuge kommen zu lassen:

```
if taskname = "XYZ"
    donotpatch = true
endif
```

1.9 Konfiguration - Requester Look

```
+-----+
| INFOLOGO (STRING)                                     (Vorgabe = $INFOLOGO) |
+-----+
```

Vorgabe Logo für Requester mit nur einem Button.

```
+-----+
| ASKLOGO (STRING)                                     (Vorgabe = $ASKLOGO) |
+-----+
```

Vorgabe Logo für Requester mit zwei Buttons.

```
+-----+
| MULTIASKLOGO (STRING)                               (Vorgabe = $MULTIASKLOGO) |
+-----+
```

Vorgabe Logo für Requester mit drei oder mehr Buttons.

```
+-----+
| LOGO (STRING) (Vorgabe = "" = verw. INFOLOGO, ASKLOGO bzw. MULTIASKLOGO) |
+-----+
```

Logo für den Requester. Hat Vorrang vor INFOLOGO, ASKLOGO, MULTIASKLOGO.

```
+-----+
| NOLOGO (BOOL)                                (Vorgabe = NO) |
+-----+
```

Kein Requesterlogo verwenden.

```
+-----+
| BACKGROUND (STRING)          (Vorgabe = "" = standard Hintergrundmuster) |
+-----+
```

Image, das als Fenster Hintergrundmuster verwendet werden soll.

```
+-----+
| BGPATTERN1PEN (FARBE)                                (Vorgabe = SHINEPEN) |
+-----+
```

Wenn man kein Image als Fenster Hintergrundmuster hat, dann wird das standard Amiga Requester Hintergrundmuster verwendet. Dieses Muster ist ein 2-Farben-Schachbrettmuster und verwendet standardmäßig SHINEPEN (weiß) als erste Farbe und BACKGROUNDPEN (grau) als zweite Farbe. Mit BGPATTERN1PEN kann man die erste Farbe ändern.

```
+-----+
| BGPATTERN2PEN (FARBE)                                (Vorgabe = BACKGROUNDPEN) |
+-----+
```

Wenn man kein Image als Fenster Hintergrundmuster hat, dann wird das standard Amiga Requester Hintergrundmuster verwendet. Dieses Muster ist ein 2-Farben-Schachbrettmuster und verwendet standardmäßig SHINEPEN (weiß) als erste Farbe und BACKGROUNDPEN (grau) als zweite Farbe. Mit BGPATTERN2PEN kann man die zweite Farbe ändern.

```
+-----+
| TEXTBOXBGPEN (FARBE)                                (Vorgabe = BACKGROUNDPEN) |
+-----+
```

Hintergrundfarbe der Box, die den Requestertext und das Logo enthält.

```
+-----+
| TEXTBOXTEXTPEN (FARBE)                                (Vorgabe = TEXTPEN) |
+-----+
```

Farbe des Requestertextes.

```
+-----+
| OTEXTBOXEDGE1PEN (FARBE)                                (Vorgabe = SHADOWPEN) |
+-----+
```

Farbe der äußeren Kanten links und oben der Box, die den Requestertext und das Logo enthält.

```
+-----+
| OTEXTBOXEDGE2PEN (FARBE)                                (Vorgabe = SHINEPEN) |
+-----+
```

Farbe der äußeren Kanten rechts und unten der Box, die den Requestertext und das Logo enthält.

```
+-----+
| ITEXTBOXEDGE1PEN (FARBE)                                (Vorgabe = <ignorieren>) |
+-----+
```

Farbe der inneren Kanten links und oben der Box, die den Requestertext und das Logo enthält. Wenn diese Variable nicht geändert wird, dann werden die inneren Kanten links und oben nicht gezeichnet und deshalb nur ein Pixel dick sein.

```
+-----+
| ITEXTBOXEDGE2PEN (FARBE)                                (Vorgabe = <ignorieren>) |
+-----+
```

Farbe der inneren Kanten rechts und unten der Box, die den Requestertext und das Logo enthält. Wenn diese Variable nicht geändert wird, dann werden die inneren Kanten rechts und unten nicht gezeichnet und deshalb nur ein Pixel dick sein.

1.10 Konfiguration - Button Look

```
+-----+
| BUTTONSTYLE (...)                                         (Vorgabe = <ignorieren>) |
+-----+
```

Setzt einige der folgenden Konfigurations Variablen auf vordefinierte Werte, je nach WERT. Mögliche WERTe sind:

```
AMITHICK
AMITHIN
WIN95A
WIN95B
KDE
QT
MONOA
MONOB
```

Um zu sehen wie die einzelnen Stile aussehen, probieren Sie sie doch einfach aus! Beachten Sie, daß andere Variablen, die von BUTTONSTYLE gesetzt werden, dennoch überschrieben werden können, indem man diese anderen Button Look Konfigurations Variablen irgendwo später im Konfigurationsskript auf andere Werte setzt.

```
+-----+
```



```
| OBUTTONEDGE1PEN (FARBE) (Vorgabe = SHINEPEN) |
+-----+
```

Farbe der äußeren Buttonkanten links und oben.

```
+-----+
| OBUTTONEDGE2PEN (FARBE) (Vorgabe = SHADOWPEN) |
+-----+
```

Farbe der äußeren Buttonkanten rechts und unten.

```
+-----+
| IBUTTONEDGE1PEN (FARBE) (Vorgabe = <ignorieren>) |
+-----+
```

Farbe der inneren Buttonkanten links und oben. Wenn diese Variable nicht geändert wird, dann werden die inneren Kanten links und oben nicht gezeichnet und deshalb nur 1 Pixel dick sein.

```
+-----+
| IBUTTONEDGE2PEN (FARBE) (Vorgabe = <ignorieren>) |
+-----+
```

Farbe der inneren Buttonkanten rechts und unten. Wenn diese Variable nicht geändert wird, dann werden die inneren Kanten rechts und unten nicht gezeichnet und deshalb nur 1 Pixel dick sein.

```
+-----+
| OBUTTONEDGE1SELPEN (FARBE) (Vorgabe = SHADOWPEN) |
+-----+
```

Farbe der äußeren Buttonkanten links und oben, wenn der Button selektiert ist (wenn Sie ihn angeklickt haben).

```
+-----+
| OBUTTONEDGE2SELPEN (FARBE) (Vorgabe = SHINEPEN) |
+-----+
```

Farbe der äußeren Buttonkanten rechts und unten, wenn der Button selektiert ist (wenn Sie ihn angeklickt haben).

```
+-----+
| IBUTTONEDGE1SELPEN (FARBE) (Vorgabe = <ignorieren>) |
+-----+
```

Farbe der inneren Buttonkanten links und oben, wenn der Button selektiert ist (wenn Sie ihn angeklickt haben). Wenn diese Variable nicht geändert wird, dann werden die inneren Kanten links und oben nicht gezeichnet und deshalb nur 1 Pixel dick sein.

Stil der Buttontexte bei hervorgehobenen Buttons. WERT kann eine Kombination der folgenden Zeichen sein:

B: BOLD (FETT)
 I: ITALIC (SCHRÄG)
 U: UNDERLINED (UNTERSTRICHEN)

SHIFTBUTTONTEXT (BOOL)	(Vorgabe = NO)
------------------------	----------------

Button Text und Image ein Pixel nach rechts und nach unten verschieben, wenn Button selektiert (angeklickt) ist.

BUTTONISPACINGX (NUMERISCH)	(Vorgabe = 4)
-----------------------------	---------------

Abstand zwischen dem linken/rechten Buttonrahmen und dem Button Text bzw. Button Image.

BUTTONISPACINGY (NUMERISCH)	(Vorgabe = 2)
-----------------------------	---------------

Minimaler Abstand zwischen dem oberen/unteren Buttonrahmen und dem Button Text.

BUTTONISPACINGY2 (NUMERISCH)	(Vorgabe = 2)
------------------------------	---------------

Minimaler Abstand zwischen dem oberen/unteren Buttonrahmen und dem Button Image.

BUTTONEXTRAWIDTH (NUMERISCH)	(Vorgabe = 8)
------------------------------	---------------

Die Mindestbreite eines Buttons errechnet sich aus der Breite des Textes + 2 x BUTTONISPACINGX + BUTTONEXTRAWIDTH. Falls Button Images verwendet werden, wird zu diesem Wert noch die Breite des Images + BIMAGETEXTSPACING dazugezählt.

BIMAGETEXTSPACING (NUMERISCH)	(Vorgabe = 4)
-------------------------------	---------------

Mindestabstand zwischen Button Image und Button Text.

Taste zum Betätigen des hervorgehobenen Buttons. Hat die gleiche Auswirkung wie das Drücken von RETURN.

```
+-----+
| CANCELKEY (STRING)                                (Vorgabe = "B") |
+-----+
```

Taste zum Betätigen des Buttons ganz rechts (Abbrechen). Hat die gleiche Auswirkung wie das Drücken von ESCAPE.

```
+-----+
| KEYQUALIFIER (..)                                (Vorgabe = LCOMMAND) |
+-----+
```

OKKEY und CANCELKEY funktionieren nur, wenn zusätzliche diese Qualifiziertaste gedrückt gehalten ist. WERT muß einer (!) der folgenden Spezialnamen sein:

```
LALT
RALT
ALT
LSHIFT
RSHIFT
SHIFT
LCOMMAND
RCOMMAND
COMMAND
CONTROL
ANY
```

ALT bedeutet, daß jede ALT Taste okay ist, SHIFT bedeutet, daß jede SHIFT Taste okay ist und COMMAND bedeutet, daß jede COMMAND Taste okay ist. ANY bedeutet, daß es vollkommen egal ist, ob und wenn ja welche Qualifiziertasten gedrückt gehalten sind.

```
+-----+
| FUNCTIONKEYS (BOOL)                                (Vorgabe = NO) |
+-----+
```

Möglichkeit, die Buttons mit den Funktionstasten (F1, F2, F3, ...) zu aktivieren.

```
+-----+
| COOLDRAGGING (BOOL)                                (Vorgabe = YES) |
+-----+
```

Wenn aktiviert, dann kann man das Requester Fenster verschieben, indem man eine beliebige Stelle im Fenster anklickt, also auch außerhalb der Titelleiste.

1.12 Konfiguration - Requester Layout

```
+-----+
| OUTERSPACINGX (NUMERISCH)                                (Vorgabe = 4) |
+-----+
```

Abstand in Pixeln zwischen Fensterrahmen und dem Rahmen der Requester Box (die den Requestertext und das Logo enthält) an der linken und an der rechten Seite.

```
+-----+
| OUTERSPACINGY (NUMERISCH)                                (Vorgabe = 4) |
+-----+
```

Abstand in Pixeln zwischen A) dem oberen Fensterrand und dem oberen Rand der Requester Box (die den Requestertext und das Logo enthält) und B) zwischen dem unteren Fensterrand und den Buttons.

```
+-----+
| INNERSPACINGX (NUMERISCH)                                (Vorgabe = 7) |
+-----+
```

Minimaler Abstand in Pixeln zwischen dem Rahmen der Requester Box und dem Requester Text/Logo auf der linken und der rechten Seite.

```
+-----+
| INNERSPACINGY (NUMERISCH)                                (Vorgabe = 7) |
+-----+
```

Minimaler Abstand in Pixeln zwischen dem Rahmen der Requester Box und dem Requester Text/Logo oben und unten.

```
+-----+
| BODYGADGETSPACING (NUMERISCH)                            (Vorgabe = 4) |
+-----+
```

Abstand in Pixeln zwischen Requester Box und Buttons.

```
+-----+
| BUTTONSPACING (NUMERISCH)                                (Vorgabe = 8) |
+-----+
```

Minimaler Abstand in Pixeln zwischen den einzelnen Buttons.

```
+-----+
| BODYLOGOSPACEING (NUMERISCH)                            (Vorgabe = 8) |
+-----+
```

Minimaler Abstand in Pixeln zwischen Requester Logo und Requester Text.

```
+-----+
| LOGOALIGN (..)                                (Vorgabe = CENTER) |
+-----+
```

Vertikale Ausrichtung des Requester Logos wenn der Requester Text höher als das Logo ist. Mögliche WERTE sind:

```
TOP      (oben)
CENTER   (mitte)
BOTTOM   (unten)
```

```
+-----+
| BODYXALIGN (..)                              (Vorgabe = CENTER) |
+-----+
```

Horizontale Ausrichtung des Requester Textes wenn die Requester Box breiter als nötig ist, das heißt, wenn die Buttonreihe eine sehr große Breite einnimmt. Mögliche WERTE sind:

```
LEFT     (links)
CENTER   (mitte)
RIGHT    (rechts)
```

```
+-----+
| BODYYALIGN (..)                              (Vorgabe = CENTER) |
+-----+
```

Vertikale Ausrichtung des Requester Textes wenn das Logo höher als der Text ist. Mögliche WERTE sind:

```
TOP      (oben)
CENTER   (mitte)
BOTTOM   (unten)
```

```
+-----+
| REQTEXTALIGN (..)                            (Vorgabe = LEFT) |
+-----+
```

Ausrichtung der Requestertext Zeilen. Mögliche WERTE sind:

```
LEFT (links)
CENTER (mitte)
RIGHT (rechts)
```

Verwechseln Sie REQTEXTALIGN nicht mit BODYXALIGN. REQTEXTALIGN bestimmt die Ausrichtung der einzelnen Text Zeilen in der imaginären Box, die den gesamten Requester Text umgibt. Mit CENTER erhält man z. B. Requester, die in etwa so aussehen:

```
+-----+
| Test Requester          |o|
+-----+
|                          |
```



```

| +-----+      Dies ist der  |
| |         |      Text eines   |
| | LOGO   | Test Requesters! |
| |         |                  |
| +-----+                  |
|
| +-----+      +-----+    |
| | OK    |      ! Abbrechen | |
| +-----+      +-----+    |
+-----+

```

```

+-----+
| BUTTONALIGN (..)                                (Vorgabe = SPREAD) |
+-----+

```

Horizontale Ausrichtung der Buttonreihe wenn mehr Platz als nötig vorhanden ist, das heißt, wenn die Requester Box breiter als die normale (minimale) Breite der Buttonreihe ist. Mögliche Werte sind:

```

LEFT : |
| +-----+ +-----+ +-----+ |
| | Eins | | Zwei | | Drei | |
| +-----+ +-----+ +-----+ |
+-----+

```

```

CENTER: |
| +-----+ +-----+ +-----+ |
| | Eins | | Zwei | | Drei | |
| +-----+ +-----+ +-----+ |
+-----+

```

```

RIGHT : |
| +-----+ +-----+ +-----+ |
| | Eins | | Zwei | | Drei | |
| +-----+ +-----+ +-----+ |
+-----+

```

```

SPREAD: |
| +-----+ +-----+ +-----+ |
| | Eins | | Zwei | | Drei | |
| +-----+ +-----+ +-----+ |
+-----+

```

```

+-----+
| SAMEBUTTONWIDTH (BOOL)                                (Vorgabe = NO) |
+-----+

```

Alle Buttons sollen gleich breit sein. Normalerweise hängt die Breite eines Buttons nur von der Breite seines Textes und seines Images ab.

```
+-----+
| REQPOS (..)                                (Vorgabe = NORMAL) |
+-----+
```

Position, an der der Requester erscheinen soll. Mögliche WERTe sind:

NORMAL: Obere linke Ecke des Bildschirms.
 CENTER: Mitte des Bildschirms.
 MOUSE : Zentriert unter dem Mauszeiger.

1.13 Konfiguration - Check Variablen

Es folgt eine Liste aller Check Variablen. Welche Art von WERT die jeweilige Check Variable erwartet steht zwischen Klammern. Bei Strings kann man ein AmigaDOS Muster (Pattern) angeben, z. B. "#?löschen#?", was auf all jene von der Check Variable "zurückgegebenen" Strings paßt, die irgendwo das Wort "löschen" enthalten (wie "löschen", "Datei löschen", "Löschen der Datei" oder "Das Löschen der Datei XY ist nicht möglich").

```
+-----+
| TASKNAME (STRING)                          |
+-----+
```

Der Taskname des Programms, welches den Requester öffnet.

```
+-----+
| PROCNAME (STRING)                          |
+-----+
```

Der Prozeßname des Programms, welches den Requester öffnet. Von Programmen erzeugte Sub-Tasks sind sehr oft keine Prozesse (ein Prozeß ist ein erweiterter Task - alle Programme die von der Workbench oder einer Shell gestartet werden sind Prozesse, - jeder Prozeß ist ein Task, aber nicht jeder Task ist ein Prozeß) und haben deshalb auch keinen Prozeßnamen.

```
+-----+
| PROGNAME (STRING)                          |
+-----+
```

Der Taskname des Programms, welches den Requester öffnet, oder, falls es keinen Tasknamen gibt, der Prozeßname.

```
+-----+
| PROGNAME2 (STRING)                         |
+-----+
```

Der Prozeßname des Programms, welches den Requester öffnet, oder, falls es sich nicht um einen Prozeß handelt oder der Prozeß keinen Namen hat,

der Taskname.

```
+-----+
| SCREENTITLE (STRING) |
+-----+
```

Der Titel des Screens auf dem der Requester erscheint.

```
+-----+
| REQTITLE (STRING) |
+-----+
```

Der Fenster-Titel des Requesters.

```
+-----+
| REQTEXT (STRING) |
+-----+
```

Der Requestertext. Zeilen sind durch Linefeed Zeichen getrennt (ASCII Code 10).

```
+-----+
| SCREENDEPTH (NUMERISCH) |
+-----+
```

Die Tiefe (Bits pro Farbe) des Screens, auf dem der Requester erscheint.
1 bedeutet 2 Farben, 4 bedeutet 16 Farben, 8 bedeutet 256 Farben, 16
bedeutet 65536 Farben, usw. (2^{DEPTH} = Anzahl Farben).

```
+-----+
| SCREENWIDTH (NUMERISCH) |
+-----+
```

Breite des Screens in Pixeln.

```
+-----+
| SCREENHEIGHT (NUMERISCH) |
+-----+
```

Höhe des Screens in Pixeln.

```
+-----+
| VISIBLEWIDTH (NUMERISCH) |
+-----+
```

Sichtbare Breite des Screens in Pixeln.

```
+-----+
| VISIBLEHEIGHT (NUMERISCH) |
+-----+
```

Sichtbare Höhe des Screens in Pixeln.

| RESOLUTIONX (NUMERISCH) |

Auflösung-Y-Wert der DrawInfo Struktur des Screens. Je niedriger dieser Wert desto höher die Auflösung.

| RESOLUTIONY (NUMERISCH) |

Auflösung-X-Wert der DrawInfo Struktur des Screens. Je niedriger dieser Wert desto höher die Auflösung.

| ASPECT (NUMERISCH) |

Kombination von RESOULTIONX und RESOLUTIONY. Zurückgegeben wird 1 für 1:1 Screens (320 x 256, 640 x 512, 1280 x 1024), 2 für 2:1 Screens (640 x 256, 1280 x 512), 4 für 4:1 Screens (1280 x 256), -2 für 1:2 Screens (320 x 512, 640 x 1024), -4 für 1:4 Screens (320 x 1024).

| NUMBUTTONS (NUMERISCH) |

Anzahl Buttons des Requesters.

1.14 DataTypes

Die ReqAttack Datatypes sind vor allem für den Preview der Images gedacht. Leider gibt es da ein paar Probleme mit dem ReqAttack AnimBrush Datatype. Er hat ein paar Bugs. Das Timing scheint nicht richtig zu funktionieren - die Animationswiedergabe scheint nicht beim ersten Frame zu beginnen :(Für jede Art von Hilfe oder Tip wäre ich sehr dankbar!

Es ist möglich, den ReqAttack AnimBrush DataType "on the fly" zu deaktivieren, indem man die LINKE ALT + LINKE UMSCHALT + CONTROL (STEUERUNG) Taste gedrückt hält, während das Image geladen wird. Der ReqAttack Brush Datatype wird "einspringen" und Sie werden ein Bild erhalten, wo alle Frames des AnimBrushes in einer Reihe gruppiert sind. Diese Prozedur wird nicht von der Shell aus funktionieren, aufgrund deren speziellen Verhaltens, wenn oben genannte Tasten gedrückt gehalten sind, während RETURN betätigt wird.

1.15 Bugs

- o Das Laden der Images ist langsam, zumindest auf meinem 68030/50er A1200 :(
- o Der ReqAttack AnimBrush Datatype funktioniert nicht richtig :(
- o ReqAttack kann gefährlich sein. Machen Sie zuerst Versuche mit nicht so gefährlichen Requestern, um zu sehen ob diese korrekt funktionieren (OK = Okay und ABBRUCH = Abbruch). Seien Sie besonders mit "Disk formattieren" Requestern vorsichtig ;-)
- o Wenn das Beantworten bestimmter Requester zu Crashes führt, stellen Sie sicher, daß Sie CyReq und/oder StackAttack verwenden.
- o Senden Sie Bug Reports an georg.steger@rolmail.net!

1.16 Trouble Shooting

- o Die AssignWedge/WBAbout Option meines XYZ Commodities funktioniert nicht mehr, wenn ReqAttack läuft:

Verwenden Sie CyReq !

- o Der Computer stürzt ab nach Beantworten von "Benötige den Datenträger XYZ" Requestern.

Verwenden Sie CyReq !

Es handelt sich hier offenbar um ein Stack-Problem der ErrorReport() Funktion aus der dos.library. Für mehr Infos siehe Geschichte von V 1.0 !

1.17 Geschichte

V 1.1
V 1.0
V 0.9

- V 1.1 (07.09.1999)
- o Neue Version von CyReq. Jetzt wird Stack Swapping auch im AutoRequest() Patch verwendet.
 - o Die EasyRequestArgs() und AutoRequest() Patches von ReqAttack, die nur aktiviert werden, wenn CyReq nicht läuft, sind jetzt mit denen von CyReq fast zu 100 % identisch, inkl. Stack Swapping usw.
 - o Neue Konfigurations Variable REQTEXTALIGN, die zum Ändern der Ausrichtung der Requestertext Zeilen dient.
 - o Neue Konfigurations Variable FUNCTIONKEYS, mit der man Button Aktivierung über Funktionstasten einschalten kann.

- o Verbesserte Grafikkarten Unterstützung dank Stephan Rupprecht. Bei HiColor und TrueColor Bildschirmmodi werden die Button Images und Requester Logos jetzt standardmäßig in derselben Farbtiefe erzeugt, so daß Farb Remapping nicht mehr notwendig ist und dadurch eine bessere Qualität erreicht wird. Bei 8 Bit Grafikkarten Screens werden die Bitmaps jetzt normalerweise im Chunky Format erzeugt, um Blinkeffekte zu vermeiden. Konfigurations Variable BITMAPFLAGS .
- o Neues Vergleichszeichen IN.
- o Neue Check Variable PROGNAME2 . Außerdem war die Beschreibung zu PROGNAME falsch.
- o Animierte Logos werden auch auf standard Amiga Screens nur mehr mit minimalen Blinkeffekten dargestellt, wenn der Screen interleavt ist.
- o Die Requester wurden mit IDCMP_CLOSEWINDOW geöffnet, obwohl die Requester Fenster kein Close Gadget besitzen. Dadurch funktionierte das Drücken der ESCAPE Taste nicht, wenn das Commodity NED benutzt wurde.
- o Neue Konfigurations Variable DONOTPATCH .
- o Neues Tool WinInfo . Hilfreich zum Herausfinden, welcher Task/Prozeß einen bestimmten Requester (oder anderes Fenster) geöffnet hat.
- o Es jetzt möglich das Ausführen der Konfigurationsdatei durch Verwenden von STOP an jeder beliebigen Stelle zu beenden,

- V 1.0 (03.08.1999)
- o Jede Menge neuer Konfigurations Variablen, um den Look von Requestern (und im speziellen seiner Buttons) noch weiter zu verbessern.
 - o Für Konfigurations Variablen, die eine Farbe erwarten, kann man nun auch einen direkten Colormap (Palette) Eintrag angeben, indem ein '#' Zeichen vorangestellt wird.
 - o Die Hintergrund Muster funktionierten unter OS 3.0 überhaupt nicht, und auch unter OS 3.1 gab es da noch ein paar Bugs.
 - o Transparenz Bug mit CyberGrahpx + SuperLayers sollte verschwunden sein. Smart Refresh Fenster, die mit LAYERS_NOBACKFILL erzeugt wurden, werden von der standard Amiga layers.library trotzdem mit Farbe 0 backgefillt, obwohl das eigentlich nicht so sein sollte. Die SuperLayers Library macht es offensichtlich richtig. Ich fand das erst heraus, nachdem jemand den Bug gemeldet hat - ich hab' keine Grafikkarte.
-

- o Die Check Variable SCREENDEPTH gab falsche Werte für HiColor und TrueColors Screens zurück. Ich verwendete DrawInfo->dri_Depth, und es stellte sich heraus, daß dieser Eintrag in der DrawInfo nie größer als 8 ist. Seltsam ...
- o Commodities Schnittstellte. Man kann ReqAttack vorübergehend deaktivieren (neu geöffnete Requester werden durch ReqAttack nicht enhanced) und beenden, indem man Exchange dazu verwendet.
- o Neue Version von CyReq. Auf manchen Systemen scheint der dos.library Funktion ErrorReport() (welche zum Öffnen des Requesters einen neuen Task erzeugt) sehr schnell der Stack auszugehen. Auf einigen Computern reichte es aus "dummertest:" im "Befehl ausführen" Fenster der Workbench einzugeben, um einen Crash zu verursachen, sobald der Benutzer den Requester, der nach der Disk "dummertest:" fragt, beantwortet hat. Das sollte jetzt nicht mehr passieren, weil CyReq jetzt Stack Swapping in seinem EasyRequestArgs() Patch benutzt. Das seltsame daran ist, daß ReqAttack schon seit jeher Stack Swapping in seinem BuildEasyRequestArgs() Patch benutzt hat (was von CyReq's EasyRequestArgs() Patch aufgerufen wird). Auf einem PC, unter dem UAE + Picasso96 läuft, reichte CyReq alleine schon aus, um die Crashes, aufgrund von Stack Overflow (glaub' ich), zu verursachen. Das ist sehr seltsam, weil CyReq sehr wenig Stack verbraucht und auch sonst so gut wie nichts macht.

Ich habe die Stelle im ROM gefunden, wo ErrorReport() seinen Task mit zu wenig Stack erzeugt, deshalb kann es sein, daß ich mal ein kleines Utility zum Patchen einer ROM Datei und/oder ein BlizKick Modul machen werde.

- o Neues Tool "QuitReqAttack". Kann dazu verwendet werden, ReqAttack von Shell Skripts oder ToolManager ähnlichen Programmen aus zu beenden.
- o Verbesserungen an den Docs.

V 0.9 (09.07.1999) o Erstes öffentliches Release.

1.18 Zukunft

- o Sound Support (Hilfe gesucht!)
- o Weiß nicht. Wenn Sie ein paar gute Ideen haben, lassen Sie es mich wissen!

1.19 Autor

ReqAttack wurde programmiert von:

Georg Steger
Hochlercher 30

I-39030 St. Johann/Ahrntal

EMail: georg.steger@rolmail.net

Wenn Sie mich kontaktieren wollen und eine Antwort erwarten, dann schicken Sie mir eine EMail. Ich werde nur EMail's beantworten. Ich hasse Snail Mail ... Meine Muttersprache ist Deutsch, Sie können aber auch in Italienisch oder Englisch schreiben.

***** ReqAttack ist Shareware!! Lesen Sie Gesetzliches ! *****

1.20 Danksagungen

- o Vielen Dank an Stephan Rupprecht für seine RTG Tips und Routinen.
- o Die unglaublich coolen animierten Requester-Logo Beispieldateien wurden von Miikka Lehto (außer das nicht so tolle "Warn", welches von mir ist) gemacht. Miikka hat verschiedene Versionen (Größe und Anzahl Farben) von jedem Logo gerendert. Damit das ReqAttack Archiv nicht zu groß wird, hab' ich aber nur jeweils eine Version beigefügt. Die restlichen Versionen werden in einem extra Archiv im Aminet erscheinen!
- o Einige der Beispiel Hintergrundmuster sind aus dem MUI Archiv entnommen. Bei den anderen kann ich mich an die Herkunft leider nicht mehr erinnern. Falls jemand sein Werk erkennt und namentlich in den Credits erwähnt werden will, oder nicht damit einverstanden ist, daß es im ReqAttack Archiv mitgeliefert wird, dann laßt es mich bitte wissen!
- o Die standard Button Images wurden von MaxonDevelop 4.0 von Tilo Kühn kopiert.
- o Das standard Ein-Button Requester Logo wurde von einem wohlbekannten aber schrecklich uncoolen Betriebssystem kopiert.

1.21 Gesetzliches

ReqAttack ist Shareware. Sie können das Programm ein paar Tage oder Wochen testen. Wenn Sie danach das Programm aber noch immer benutzen, müssen Sie die Shareware Gebühr bezahlen. Alles von 2000 Lire (2 DM, 1 US\$) aufwärts bis zu 5000 Lire (5 DM, 3 US\$) ist okay. Wenn möglich bitte keine anderen Währungen. Eventuell akzeptiere ich auch ein kleines Geschenk anstelle des Geldes, aber vorher fragen (EMail)!!! Meine Adresse:

Georg Steger
Hochlercher 30

I-39030 St. Johann/Ahrntal - ITALY

EMail: georg.steger@rolmail.net

Wichtig:

- o Ich kann für keinen Schaden verantwortlich gemacht werden, der eventuell bei der Benutzung von ReqAttack entstehen könnte.
- o Ich werde keine Programme oder Programm Updates per Post verschicken!!! Sorry, aber sehen Sie sich noch einmal die extrem niedrige Shareware Gebühr an - das können Sie einfach nicht von mir erwarten.
- o Noch Fragen? Schicken Sie mir eine EMail! Ich werde nur per EMail antworten. Briefe mit der (Schnecken-) Post zu verschicken ist zu viel Arbeit!

1.22 Abschließend

Viel Spaß ;-)
