

in

Georg Steger

COLLABORATORS

	<i>TITLE :</i> in		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Georg Steger	July 10, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	in	1
1.1	ReqAttack Tools	1
1.2	RAIM_Convert	1
1.3	RAIM_SetAttributes	3
1.4	RAIM_GetInfo	3
1.5	RAIM_Anim2Brush	4
1.6	RAIM_Brush2Anim	4
1.7	RAIM_Anim2IFF	5
1.8	WinInfo	7
1.9	CyReq	8
1.10	QuitReqAttack	9

Chapter 1

in

1.1 ReqAttack Tools

ReqAttack Tools - (C) 1999 by Georg Steger

```
RAIM_Convert
RAIM_SetAttributes
RAIM_GetInfo
RAIM_Anim2Brush
RAIM_Brush2Anim
RAIM_Anim2IFF
WinInfo
CyReq
QuitReqAttack
```

1.2 RAIM_Convert

Mit RAIM_Convert können IFF Bilder oder jedes andere Bildformat, für das ein Datatype installiert ist, in ReqAttack's eigenes Imageformat konvertiert werden, welches das einzige Format ist, das ReqAttack unterstützt. Dieses Programm kann nur von der Shell aus gestartet werden. Es hat folgende Befehlsschablone:

```
SOURCE/A,DEST/A,FRAMEWIDTH=FW/N/K,FRAMEHEIGHT=FH/N/K,FRAMES=F/N/K,
FRAMESPERROW=FPR/N/K,ANIMSPEED=AS/N/K,NOTRANSP=NT/S,DONTPACK=DP/S,
PINGPONG=PP/S
```

Hauptargumente, die sowohl für Brushes als auch Animbrushes erforderlich sind:

SOURCE: Name der Image Quelldatei.

DEST : Name der Image Zieldatei.

Argumente die zusätzlich bei Animbrushes erforderlich sind. Beachten Sie, daß in der Image Quelldatei alle Frames (Einzelbilder) auf die folgende Weise gruppiert sein müssen:

```
+---+---+---+---+---+
! A | B | C | D | E |
+---+---+---+---+---+
! F | G | H | I | J |
+---+---+---+---+---+
! K | L | M | N | O |
+---+---+---+---+---+
! P | Q |
+---+---+
```

Das erste Frame (A) muß an Pixel Position (0,0) beginnen. Zwischen den einzelnen Frames darf kein Freiraum sein.

FRAMEWIDTH : Breite eines Frames in Pixeln.

FRAMEHEIGHT : Höhe eines Frames in Pixeln.

FRAMES : Gesamtanzahl an Frames. Im obigen Beispiel sind es 17.

FRAMESPERROW: Anzahl Frames pro Reihe. Im obigen Beispiel haben wir 5 Frames pro Reihe. (Die letzte Reihe ist irrelevant).

ANIMSPEED : Wartepause in 1/50 Sekunden zwischen den Frames. Kann später noch mit dem Tool RAIM_SetAttributes geändert werden.

Die folgenden Argumente sind optional, müssen also nicht angegeben werden:

NOTRANSP: Keine Transparenz der Farbe 0. Die Farbe 0 wird normalerweise nicht remapped. Mit NOTRANSP wird sie das doch. Für weitere Informationen siehe RAIM_SetAttributes Dokumentation.

PINGPONG: Normalerweise beginnt nach dem Anzeigen des letzten Frames die Animationswiedergabe erneut beim ersten Frame. Mit dieser Option wird die Abspielrichtung beim Erreichen des letzten Frames auf rückwärts und dann beim Erreichen des ersten Frames wieder auf vorwärts umgestellt. Für weitere Informationen siehe RAIM_SetAttributes Dokumentation.

DONTPACK: Normalerweise komprimiert RAIM_Convert die Image Daten mit einem recht primitiven Algorithmus, der sich ByteRun1 nennt, und der dem des IFF ILBM Formats entspricht. In einigen Fällen kann es vorkommen, daß die Image Datei mit Komprimierung größer wird als ohne. RAIM_Convert überprüft das aber nicht. Mit dieser Option wird die Komprimierung ausgeschaltet.

1.3 RAIM_SetAttributes

Mit RAIM_SetAttributes können einige Eigenschaften einer [R]eq[A]ttack [I]mage Datei geändert werden. Dieses Programm kann nur von der Shell aus gestartet werden. Es hat folgende Befehlsschablone:

```
FILE/A, TRANSP=TR/S, NOTRANSP=NOTR/S, PINGPONG=PP/S, NOPINGPONG=NOPP/S,
ANIMSPEED=AS/N/K
```

FILE : Name der Image Datei.

TRANSP : Schaltet die Transparenz von Farbe 0 ein. Dies bedeutet, daß egal welche Palette die Farbe 0 hat, sie dennoch nicht remapped wird. Wenn z. B. die Palette (RGB Wert) der Farbe 0 des Images schwarz ist, und die Palette der Farbe 0 des Screens in welchem der Requester erscheint grau ist, dann werden jene Teile des Images, die mit Farbe 0 gezeichnet sind, grau erscheinen.

NOTRANSP : Schaltet die Transparenz von Farbe 0 aus. Teile des Images, die mit Farbe 0 gezeichnet sind, werden durch jene Farbe des Screens ersetzt, deren Palette am ehesten der Palette der Farbe 0 des Images entspricht.

PINGPONG : Schaltet Ping-Pong Animation ein. Wenn Sie zum Beispiel einen AnimBrush mit 10 Frames haben, dann wird die Animation auf folgende Weise abgespielt:

```
1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10 9 8 7
```

NOPINGPONG: Schaltet Ping-Pong Animation aus. Nachdem das letzte Frame angezeigt wurde, beginnt die Wiedergabe erneut beim ersten Frame:

```
1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1
```

ANIMSPEED : Wartepause in 1/50 Sekunden zwischen den Frames.

1.4 RAIM_GetInfo

Mit diesem Tool kann man einige Informationen über die angegebene ReqAttack Image Datei abfragen. Das Programm kann nur von der Shell aus gestartet werden. Es hat folgende Befehlsschablone:

```
FILE/A, WIDTH=W/S, HEIGHT=H/S, FRAMEWIDTH=FW/S, FRAMEHEIGHT=FH/S,
FRAMES=FR/S, COLORS=CO/S, PLANES=PL/S, ANIMSPEED=AS/S, FLAGS=FL/S,
ANIMFLAGS=AF/S
```

Wenn nur das FILE Argument angegeben wird, dann werden alle Informationen ausgegeben!

FILE : Name der Image Datei.

WIDTH : Ausgabe der Breite des Images. Bei ReqAttack Animbrushes entspricht dies FRAMEWIDTH x FRAMES!

HEIGHT : Ausgabe der Höhe des Images.

FRAMEWIDTH : Ausgabe der Breite eines Frames. Wenn die Image Datei kein ReqAttack AnimBrush sondern ein ReqAttack Brush ist, dann wird die Breite des Images ausgegeben.

FRAMEHEIGHT: Ausgabe der Höhe eines Frames. Entspricht immer HEIGHT!

FRAMES : Ausgabe der Anzahl an Frames wenn die Datei ein ReqAttack AnimBrush ist, "1" im Falle eines ReqAttack Brushes.

COLORS : Ausgabe der Anzahl an Farben der Image Datei.

PLANES : Ausgabe der Anzahl an Planes (Bits pro Farbe) der Image Datei.

ANIMSPEED : Ausgabe der Animationsgeschwindigkeit (in 1/50 Sekunden) oder 0 wenn die Image Datei ein ReqAttack Brush ist und kein ReqAttack AnimBrush.

FLAGS : Ausgabe der Image Flags (PACKED, NOTTRANSPARENCY).

ANIMFLAGS : Ausgabe der Animation Flags (PINGPONG).

1.5 RAIM_Anim2Brush

Mit diesem Tool kann ein ReqAttack AnimBrush in einen ReqAttack Brush konvertiert werden. Der Brush wird alle Frames in einer Reihe zusammengruppiert haben. Dieses Programm kann nur von der Shell aus benutzt werden. Es hat folgende Befehlsschablone:

```
FILE/A
```

FILE: Name der Image Datei.

Sie werden dieses Programm wahrscheinlich nie brauchen, wegen eines Spezialfeatures des ReqAttac k AnimBrush Datatypes, das Ihnen erlaubt den Datatype "on the fly" zu deaktivieren, um den ReqAttack Brush Datatype einspringen lassen zu können. Alles was dazu notwendig ist, die Tasten LINKE ALT + LINKE SHIFT + CONTROL gedrückt zu halten während das Image geladen wird (von Multiview oder irgendeinem anderen Programm, das die Datatypes zum Laden von Bildern verwendet). Leider funktioniert das nicht von Shells aus, aufgrund deren speziellen Verhaltens wenn oben genannte Tasten bei Betätigung von RETURN gedrückt gehalten sind.

Mit Programmen, die nur Bild (also keine Animation) Datatypes laden, sollte es sowieso keinerlei Probleme geben, weil diese Programme Animationsdatatypes automatisch ignorieren (sollten), es sei denn diese Programme kontrollieren zuert die GID (Group ID) des Datatypes.

1.6 RAIM_Brush2Anim

Mit diesem Tool kann eine ReqAttack Image Datei, die vorher von einem ReqAttack AnimBrush in einen ReqAttack Brush konvertiert wurde (mit dem Tool

```
RAIM_Anim2Brush
```

) zurück in einen ReqAttack AnimBrush konvertiert werden.

Dieses Programm kann nur von der Shell aus benutzt werden. Es hat folgende Befehlsschablone:

```
FILE/A
```

FILE: Name der Image Datei.

1.7 RAIM_Anim2IFF

Mit diesem Tool können entweder einzelne Frames eines ReqAttack AnimBrushes in IFF Bilder oder alle Frames eines ReqAttack AnimBrushes in ein IFF Bild, das alle Frames enthält, konvertiert werden. Dieses Programm kann nur von der Shell aus benutzt werden. Es hat folgende Befehlsschablone:

```
SOURCE/A,DEST/A,FRAMESPERROW=FPR/N/K,STARTFRAME=SF/N/K,ENDFRAME=EF/N/K,
FRAME=FR/N/K,ALLFRAMES=AF/S,VERBOSE=V/S
```

SOURCE : Name der ReqAttack AnimBrush Datei.

DEST : Name der IFF Zieldatei(en). Wenn mehrere Frames in IFF Bilder konvertiert werden sollen, dann muß ~irgendwo im Dateinamen "%ld" vorkommen. Dies wird durch die entsprechende Framenummer ersetzt. Zum Beispiel ergibt "Frame%ld.iff" folgende Dateinamen:

```
"Frame1.iff"
"Frame2.iff"
"Frame3.iff"
...
"Frame10.iff"
"Frame11.iff"
...
```

Anstelle von "%ld" kann man auch "%02ld", "%03ld", "%04ld", ... verwenden, um führende Nullen zu erhalten:

```
"Frame%02ld.iff" | "Frame%03ld.iff" | "Frame%04ld.iff"
-----+-----+-----
"Frame01.iff"   | "Frame001,iff"   | "Frame0001.iff"
"Frame02.iff"   | "Frame002.iff"   | "Frame0002.iff"
"Frame03.iff"   | "Frame003.iff"   | "Frame0003.iff"
...             | ...             | ...
"Frame10.iff"   | "Frame010.iff"   | "Frame0010.iff"
...             | ...             | ...
```

FRAMESPERROW: Wenn sie den ReqAttack AnimBrush in eine einzige Datei konver-

tieren wollen, die alle Frames enthält, dann können Sie mit diesem Argument bestimmen, wie viele Frame pro Reihe Sie in IFF Bild gerne hätten. Normalerweise berechnet RAIM_Anim2IFF selbst einen "guten" Wert dafür, so daß in etwa gleich viele Frames pro Reihe wie Frames pro Spalte herauskommen.

- STARTFRAME : Wenn einzelne Frames in IFF Bilder konvertiert werden sollen, dann kann mit diesem Argument das Startframe angegeben werden (Vorgabe = 1 = erstes Frame).
- ENDFRAME : Mit diesem Argument kann das letzte zu konvertierende Frame bestimmt werden (Vorgabe = letztes Frame).
- FRAME : Entspricht der Angabe des gleichen Wertes für STARTFRAME und ENDFRAME.
- ALLFRAMES : Entspricht der Angabe von 1 für STARTFRAME und <ANZAHL FRAMES> für ENDFRAME.
- VERBOSE : Kurze Ausgabe nach jedem konvertierten Frame.

Durch Angabe von STARTFRAME, ENDFRAME, FRAME oder ALLFRAMES wird das Tool in den "Einzelne Frames Modus" geschaltet, d. h. eine IFF Datei pro Frame wird erzeugt. Wenn keine dieser Argumente angegeben wird, dann wird das Programm alle Frames in ein einziges IFF Bild konvertieren. Beispiele:

MeinImage ist ein ReqAttack AnimBrush mit 10 Frames.

```
RAIM_Anim2IFF MeinImage MeinImage.iff
... erzeugt das IFF Bild "MeinImage.iff" mit allen Frames zusammen-
gruppiert.
```

```
RAIM_Anim2IFF MeinImage MeinImage%ld.iff ALLFRAMES
... erzeugt 10 IFF Bilder (eines pro Frame) namens "MeinImage1.iff",
"MeinImage2.iff", ..., "MeinImage10.iff".
```

```
RAIM_Anim2IFF MeinImage MeinImage%ld.iff STARTFRAME 5 ENDFRAME 8
... erzeugt 4 IFF Bilder aus den Frames 5 - 8: "MeinImage5.iff",
"MeinImage6.iff", "MeinImage7.iff", "MeinImage8.iff".
```

```
RAIM_Anim2IFF MeinImage MeinImage.iff STARTFRAME 6 ENDFRAME 6
... erzeugt ein IFF Bild namens "MeinImage.iff" aus dem Frame 6.
```

```
RAIM_Anim2IFF MeinImage MeinImage.iff FRAME 6
... macht genau das gleiche.
```

```
RAIM_Anim2IFF MeinImage MeinImage%ld.iff ENDFRAME 3
... erzeugt 3 IFF Bilder aus den Frames 1 - 3: "MeinImage1.iff",
"MeinImage2.iff", "MeinImage3.iff".
```

```
RAIM_Anim2IFF MeinImage MeinImage%02ld.iff STARTFRAME 8
... erzeugt 3 IFF Bilder aus den Frames 8 - 10: "MeinImage08.iff",
"MeinImage09.iff", "MeinImage10.iff".
```

1.8 WinInfo

Mit diesem Tool kann man herausfinden, welchem Task/Prozeß ein bestimmtes Fenster (z. B. ein Requester) gehört. In den allermeisten Fällen entspricht dies dem Task/Prozeß, der das Fenster geöffnet hat. Es gibt aber einige besondere Hacks/Commodities, wie z. B. ToolsDaemon, die den UserPort einiger Fenster verändern. Dann kann es sein, daß die von WinInfo angezeigten Task/Prozeßnamen seltsam erscheinen. So wird bei Workbench Fenstern etwa "Tools-Daemon" anstelle von "Workbench" angezeigt, wenn ToolsDaemon läuft. Ein anderes Problem sind Fenster, die über keinen Userport verfügen, weil sie keine IDCMP Nachrichten empfangen wollen. Hier kann WinInfo nicht herausfinden, wem das Fenster gehört. Bei Requester Fenstern gibt es aber diese Probleme nicht, und deshalb kann hier WinInfo eine große Hilfe sein, um z. B. Requestern von bestimmten Programmen einen besonderen Look zu geben, indem man in der ReqAttack Konfigurationsdatei entsprechende Bedingungen einbaut.

WinInfo kann man zwar direkt aus einer Shell starten, am besten ist es aber, wenn man es zusammen mit einem Commodity verwendet, welches einem erlaubt externe Programme durch Drücken einer Funktionstaste oder eines Hotkeys zu starten. Nach dem Start kann man mit dem Mauszeiger über ein beliebiges Fenster fahren, welches von WinInfo durch ein scrollendes Rechteck markiert wird. Zu dem Fenster unter dem Mauszeiger wird eine Infobox angezeigt. Zum Beenden von WinInfo reicht es aus, die linke Maustaste zu drücken. Die Befehlsschablone von WinInfo sieht wie folgt aus:

```
FPS/N/K,HIDDENAREA=HA/N/K,LINEPATTERN1=LP1/K,LINEPATTERN2=LP2/K,
LINECOL1=LC1/K,LINECOL2=LC2/K,HLINECOL1=HLC1/K,HLINECOL2=HLC2/K,
INFOBGCOL1=IBGC1/K,INFOBGCOL2=IBGC2/K,LABELCOL=LC/K,TEXTCOL=TC/K,
INFOBORCOL=IBC/K,INFOTITLECOL=ITK/K
```

FPS : Geschwindigkeit in Bildern pro Sekunde, mit der das Rechteck-Linienmuster gescrollt wird.

HIDDENAREA : Aussehen des scrollenden Rechtecks in verdeckten Bereichen des Fensters, welches sich unter dem Mauszeiger befindet:

```
0: Unsichtbar
1: Sichtbar
2: Sichtbar aber mit alternativen Farben/Linienmuster
```

LINEPATTERN1: Normales 16 Pixel Linienmuster (0xFF00 = 1111111100000000).

LINEPATTERN2: Alternatives 16 Pixel Linienmuster (für verdeckte Bereiche).

LINECOL1 : Farbe 1 im normalen Linienmuster (wo Bits gesetzt sind).

LINECOL2 : Farbe 2 im normalen Linienmuster (wo Bits gelöscht sind).

HLINECOL1 : Farbe 1 im alternativen Linienmuster.

HLINECOL2 : Farbe 2 im alternativen Linienmuster.

INFOBGCOL1 : Hintergrundfarbe 1 der Info Box.

INFOBGCOL2 : Hintergrundfarbe 2 der Info Box.

LABELCOL : Farbe der Label Texte.
TEXTCOL : Farbe für Task/Prozeßnamen.
INFOBORCOL : Farbe des Infobox Rahmens.
INFOTITLECOL: Farbe des Infobox Titels.

Bei Argumenten, die eine Farbe erwarten, kann man entweder einen 24 Bit RGB Wert (am besten im HEX Format, z. B. 0xFFFF00 für gelb), oder einen DrawInfo Pen Namen (SHINEPEN, SHADOWPEN, FILLPEN, FILLTEXTPEN, TEXTPEN, ...) angeben.

1.9 CyReq

CyReq lenkt EasyRequestArgs() Funktionsaufrufe nach:

```
BuildEasyRequestArgs()  
SysReqHandler()  
FreeSysRequest()
```

und AutoRequest() Funktionsaufrufe nach:

```
BuildSysRequest()  
SysReqHandler()  
FreeSysRequest()
```

um. Das wird deshalb gemacht weil EasyRequestArgs() und AutoRequest() FreeSysRequest() direkt aufrufen, d. h. nicht über den LVO. Deshalb kann man solche FreeSysRequest Aufrufe (die CloseWindow(), usw. aufrufen) nicht abfangen. Mit CyReq ist das möglich.

Man kann die Funktionalität von CyReq zwar auch direkt in das Patchprogramm (ReqAttack, PowerWindows) einbauen, aber dann gibt es das Problem, das bestimmte andere Programme, die Requester ersetzen oder verbessern nicht mehr funktionieren, es sei denn diese Programme installieren ihre Patches (normalerweise ein EasyRequestArgs() Patch) nach ReqAttack, PowerWindows, ...

Indem man CyReq irgendwo nach "SetPatch" in die Startup-Sequence einträgt, sind diese Umlenkungs Patches in den anderen Programmen (ReqAttack, PowerWindows) nicht mehr notwendig, weil sie ja bereits von CyReq gemacht werden. ReqAttack erkennt CyReq automatisch, bei alten (öffentlichen) Versionen von PowerWindows muß die Option "Requester nicht patchen" im Einstellungsprogramm aktiviert werden.

Nachdem CyReq installiert ist, sollte es vollkommen egal sein, wann ReqAttack, PowerWindows, ... gestartet werden. AssignWedge Tools und ähnliche Programme sollten immer funktionieren, egal, ob sie vor oder nach ReqAttack, PowerWindows, ... gestartet wurden.

Es wird dringend angeraten, CyReq zu installieren, unter anderem deshalb, weil CyReq in seinem EasyRequestArgs() Patch Stack Swapping benutzt und damit hoffentlich die auf einigen Systemen auftretenden Crashes nach Beantworten von "Benötige den Datenträger XYZ ..." Requestern verhindert. Mehr dazu steht in der ReqAttack Geschichte zu V 1.0!

CyReq kann mit dem ReqAttack Installations Skript installiert werden.

1.10 QuitReqAttack

Dieses Programm sendet ReqAttack ein CONTROL C Signal (Break), welches ReqAttack dazuführt, sich zu beenden bzw. das zu versuchen. Wenn ReqAttack nicht läuft, dann gibt dieses Shell Programm WARN (5) als primären Rückgabewert (\$RC) zurück.
