

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	Introduction	1
1.3	Features	2
1.4	Installation	2
1.5	Usage	3
1.6	Configuration	4
1.7	Configuration Items	7
1.8	Configuration - General Settings	12
1.9	Configuration - Requester look	14
1.10	Configuration - Button look	16
1.11	Configuration - Requester handling	21
1.12	Configuration - Requester layout	22
1.13	Configuration - Check items	25
1.14	Datatypes	27
1.15	Bugs	28
1.16	Trouble Shooting	28
1.17	History	28
1.18	Future	31
1.19	Author	31
1.20	Credits	31
1.21	Legal Stuff	32
1.22	Final Notes	32

Chapter 1

in

1.1 ReqAttack Tools

ReqAttack 1.0 - (C) 1999 by Georg Steger

Introduction

Trouble Shooting

Features

History

Installation

Future

Usage

Configuration

Author

Tools

Credits

Datatypes

Legal Stuff

Bugs

Final Notes

1.2 Introduction

ReqAttack is a little program to improve the look and handling of requesters ↔

of all programs which use the standard system requester functions (EasyRequestArgs, AutoRequest, BuildEasyRequestArgs, BuildSysRequest) for that purpose.

```
***** ReqAttack is Shareware!! Read
        Legal Stuff
        ! *****
```

1.3 Features

- o Configurable requester logos which can also be animated.
- o Configurable button images.
- o Configurable button look.
- o Configurable requester background pattern.
- o Keyboard control. Cycling through buttons with TAB, SHIFT TAB and CURSOR keys. Activation of highlighted gadget with RETURN and another configurable key (default LCOMMAND v). Activation of rightmost gadget with ESCAPE and another configurable key (default LCOMMAND b). Activation of buttons by keyboard can be aborted by pressing SHIFT or ESCAPE before releasing the button activation key.
- o Possibility to drag the requesters around by clicking anywhere in the window.
- o Possibility to make the requesters open in the center of the screen or under the mouse pointer.
- o Very special conditional configuration.
- o It's possible to configure almost everything, like spacing between the single elements of the requester and it's alignment, the color of the highlight rectangle and much more.
- o Commodities Interface

1.4 Installation

Double click the Install icon to install ReqAttack on your harddisk. If you encounter some installation problems (Installer aborts with some error message) make sure to have the latest Installer version (>= 43.3) which you can find on Aminet.

ReqAttack will not be added to your Startup-Sequence, User-Startup or WB-Startup. Drag the ReqAttack icon to your WBStartup drawer after installation if you want ReqAttack to be started during boot. If you were already using an older (non public beta) version of ReqAttack then the installation script will replace a possible old executable in your WBStartup drawer as well.

If you want to change the location of the ReqAttack drawer after installation change the file "S:ReqAttack.installinfo" with a text editor to reflect the new path. If you don't do this future updates of ReqAttack may cause problems during installation!

If you want to uninstall ReqAttack you must do this by hand, because there is actually no Deinstallation script:

Main program, tools, docs and images:

- o Delete ReqAttack drawer.
- o Delete ReqAttack program in WBStartup drawer if you have dragged/copied it there as well.

ReqAttack Configuration file:

- o Delete "ENVARC:ReqAttack.prefs"

ReqAttack install information file:

- o Delete "S:ReqAttack.installinfo"

ReqAttack Datatypes:

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

CyReq (may be needed by other programs like PowerWindows!!):

- o Delete "SYS:C/CyReq"
- o Remove the "CyReq" line from "S:Startup-Sequence"

1.5 Usage

ReqAttack can be started from Shell and from Workbench. The following Shell arguments/Workbench Tooltypes can be used:

HOMEDIR: Can be used to specify the current directory which ReqAttack uses while it is running. By doing so you can use relative image filenames even when having dragged ReqAttack to the WBStartup drawer where the normal current directory for ReqAttack would otherwise be "SYS:WBStartup".

The installation script automatically sets the HOMEDIR Workbench Tooltype to the destination directory you have chosen for ReqAttack.

CYREQ : Old versions of CyReq cannot be detected automatically by ReqAttack. The new version which comes with this archive and which you can have installed automatically by the Installation script, if you wish, makes autodetection possible. If for some reason you want to keep using an old version of CyReq you must activate this option, otherwise CyReq will not have any effect.

This option of course only makes sense if you are using CyReq in your Startup-Sequence. For more information read the CyReq doc .

To quit ReqAttack send it a break (CONTROL C) or start the program again. Another possibility is to use Exchange, because since version 1.0 ReqAttack has a Commodities interface.

If quitting fails you will only see a screen flash and ReqAttack will keep running. This can happen if other programs have patched OS functions which ReqAttack patched before them. In such a case you should be able to make ReqAttack quit by first quitting this other program(s).

It's also possible to disable ReqAttack temporarily by clicking on "Disable" in Exchange. New opened requesters will then not be enhanced by ReqAttack anymore until you enable ReqAttack again.

1.6 Configuration

Quick links:
Configuration Items

Check items

Unfortunately there's no configuration program and it's uncertain whether there will ever be one as configuration is somehow special. Another reason is that I really hate writing configuration programs ...

There are two configuration files both named "ReqAttack.prefs". One is in the ENV: directory which is usually in RAM Disk and the other in the ENVARC: directory which is usually on your harddisk's boot volume. The fact that there are two configuration files is nothing strange. Many other programs as for example all standard Workbench preferences programs work the same way. The difference is that these programs have a GUI and you don't notice it.

The "ReqAttack.prefs" file in ENVARC: is the more important one. Editing it is the same as clicking on the SAVE button of a Workbench prefs program. During boot this file will be copied to ENV: by Startup-Sequence. So initially, after booting, in ENV: you have an exact copy of the saved configuration file.

Editing the "ReqAttack.prefs" file in ENV: has the same effect as clicking on the USE button of a Workbench prefs program. ReqAttack reads the configuration from ENV: when it is started and each time this file in ENV: is changed, that is when you save it with your editor.

So to configure ReqAttack you will mainly have to edit ENV:ReqAttack.prefs. If you like your actual configuration and want to save it permanently - the files in ENV: are lost after a reset - simply copy ENV:ReqAttack.prefs to ENVARC:

The configuration file is a simple text file. Lines starting with a ";" character are treated as comments and are therefore ignored. There may be spaces or tabs before the ";".

The special thing about ReqAttack's configuration file is that it is like a script which is executed every time a requester is opened. You can have conditions in it and this makes it almost look like a programming language with a similiar syntax as BASIC. Before looking on how to use conditions let's have a look at a normal line which is used to set the value of a configuration item:

```
ITEM = VALUE
```

There must be at least one space or tab character before and after the "="! VALUE is always a constant and can either be a string, a number or a special name depending on ITEM. Numeric values can also be specified in hexadecimal format by using "0x" or "0X", for example 0x100. Strings should be enclosed in quotation marks. You must do it, when the string contains spaces. Items are not case sensitive. You are even allowed to insert underscore characters ("_") to make long item names more readable.

List of all configuration items

Now let's look at how to put conditions into the configuration file. This is ←

very simple as you can see in the following example:

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

In this example the configuration item "LOGO" will be set to "cool_wb_logo" only if the program who opens the requester is the Workbench. The "logo =" line is indented only to make everything more readable. Now we want to extend the configuration file so that all requesters which are *not* opened by Workbench shall get the logo "standard_logo". We could write the config file like this:

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

This works, but it's better to write it like this:

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


Of course you can have more than one line between IF and ENDIF (and between IF and ELSE - and between ELSE and ENDIF). Nesting is possible as well:

```

if taskname = "Workbench"
    logo = "cool_wb_logo"
    highlightpen = 0xFFFF00
else
    logo = "standard_logo"
    if screendepth < 2
        highlightpen = SHADOWPEN
    else
        highlightpen = 0xFF0000
    endif
    reqpos = MOUSE
endif

```

Thus any condition must be written like this:

```

IF CHECKITEM OPERATION VALUE
    ;Lines that are executed if condition is TRUE.
ELSE
    ;Lines that are executed if condition is NOT TRUE.
ENDIF

```

ELSE and the lines between it and ENDIF are optional. There must be at least one space or tab before and after the OPERATION! Available operations are:

= Condition is TRUE if CHECKITEM equals VALUE. Instead of "=" you can also write "==" like in C.

<> Condition is TRUE if CHECKITEM does not equal VALUE. Instead of "<>" you can also write "!=" like in C.

< Condition is TRUE if CHECKITEM is lower than VALUE.

> Condition is TRUE if CHECKITEM is greater than VALUE.

<= Condition is TRUE if CHECKITEM is lower or equal VALUE.

>= Condition is TRUE if CHECKITEM is greater or equal VALUE.

IN Condition is TRUE if CHECKITEM equals one of the comma separated sub VALUES. When using IN, the VALUE must be enclosed in quotation marks! Example:

```

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

```

This condition is TRUE, if the screen depth is either 2, 4, 6 or 8 Bit. For CHECKITEMS which return a string it is of course possible to use an AmiggoDOS patterns in each sub VALUE:

```

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

```

List of all check items
 Since version 1.1 it is possible to stop the execution of the configuration ←

script at any time by using the command STOP. In some situations this can be clearer and shorter than using deeply nested conditions. So instead of ...

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

... you can also write:

```
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
```

Finally some important notes about errors (syntax errors, IF without ENDIF, etc.) in the configuration file. When ReqAttack is started and there is an error in the configuration file then it will quit and display an error message. If ReqAttack does not report any error when started and you later change the settings by editing "ENV:ReqAttack.prefs" then errors in the configuration files will not cause ReqAttack to quit. Instead you will only see a screen flash and ReqAttack will fall back to the previous (working) configuration file!

1.7 Configuration Items

Following is a list of all available configuration items

. By clicking on the links you get more information like the possible values etc. The required type of value will be shown between brackets.

For BOOL value types you can write "YES", "TRUE", "ON" (positive) or "NO", "FALSE", "OFF" (negative).

For items which expect an image file name you can also tell ReqAttack to use one of the built in standard images by writing "\$INFOLOGO", "\$ASKLOGO", "\$MULTIASKLOGO", "\$OKIMAGE", "\$CANCELIMAGE" or "\$THIRDIMAGE". For external images please note that ReqAttack only supports it's own image format. Use the utility RAIM_Convert for conversion purposes and read it's doc.

For items which expect a color you can either use a 24 Bit RGB value (by writing it's numeric value in dec or hex format), a direct colormap pen number (by putting a '#' char in front of the numeric value: #0 for example will pick the first color of the screen's palette) or one of the following special names which tell ReqAttack to use the corresponding standard Draw-Info pen of the screen on which the requester pops up:

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

? AmigaGuide Datatype cannot display these colors

If you want to use a RGB value you should write it in hex format. For a yellow color you could for example write "0xFFFF00".

GENERAL SETTINGS

IMAGECACHES

How many images to cache in memory

BITMAPFLAGS

Especially interesting for GFX card owners

DONOTPATCH

Requester shall not be patched by ReqAttack

REQUESTER LOOK

INFOLOGO
Default logo for requesters with one button

ASKLOGO
Default logo for requesters with two buttons

MULTIASKLOGO
Default logo for requesters with three or more buttons

LOGO
Logo to use

NOLOGO
No logo please

BACKGROUND
Background pattern image

BGPATTERN1PEN
Background pattern color 1

BGPATTERN2PEN
Background pattern color 2

TEXTBOXBGPEN
Background color of box containing the requester text

TEXTBOXTEXTPEN
Color to use for requester text

OTEXTBOXEDGE1PEN
Color of outer left and top edges of text box

OTEXTBOXEDGE2PEN
Color of outer right and bottom edges of text box

ITEXTBOXEDGE1PEN
Color of inner left and top edges of text box

ITEXTBOXEDGE2PEN
Color of inner right and bottom edges of text box

BUTTON LOOK

BUTTONSTYLE
Predefined button styles

OBUTTONEDGE1PEN
Color of outer left and top button edges

OBUTTONEDGE2PEN
Color of outer right and bottom button edges

IBUTTONEDGE1PEN
Color of inner left and top button edges

IBUTTONEDGE2PEN
Color of inner right and bottom button edges

OBUTTONEDGE1SELPEN
Color of outer left/top button edges (selected)

OBUTTONEDGE2SELPEN
Color of outer right/bottom button edges (selected)

IBUTTONEDGE1SELPEN
Color of inner left/top button edges (selected)

IBUTTONEDGE2SELPEN
Color of inner right/bottom button edges (selected)

OBUTTONEDGE1HIPEN
Color of outer left/top button edges (highlighted)

OBUTTONEDGE2HIPEN
Color of outer right/bottom button edges (highlighted)

IBUTTONEDGE1HIPEN
Color of inner left/top button edges (highlighted)

IBUTTONEDGE2HIPEN
Color of inner right/bottom button edges (highlighted)

BUTTONBGPEN
Button background color

BUTTONBGSELPEN
Button background color when selected

BUTTONBGHIPEN
Button background color when highlighted

BUTTONTEXTPEN
Button text color

BUTTONTEXTSELPEN
Button text color when selected

BUTTONTEXTHIPEN
Button text color when highlighted

BUTTONTEXTSTYLE
Button text style

BUTTONTEXTSELSTYLE
Button text style when selected

BUTTONTEXTHISTYLE
Button text style when highlighted

SHIFTBUTTONTEXT
Scroll button text (and image) 1 pixel when selected

BUTTONISPACINGX
Horiz. space between button frame and text/image

BUTTONISPACINGY
Min. vert. space between button frame and text

BUTTONISPACINGY2
Min. vert. space between button frame and image

BUTTONEXTRAWIDTH
Will be added to the minimum width of a button

BIMAGETEXTSPACING
Min. space between button image and text

HILIGHTPEN
Color of hilight rectangle

HITHICKX
Horizontal thickness of hilight rectangle

HITHICKY
Vertical thickness of hilight rectangle

OKIMAGE
Image for leftmost (ok) button

CANCELIMAGE
Image for rightmost (cancel) button

THIRDIMAGE
Image for any other button

NOBUTTONIMAGES
No button images at all

REQUESTER HANDLING

OKKEY
Shortcut for leftmost (ok) button

CANCELKEY
Shortcut for rightmost (cancel) button

KEYQUALIFIER
Keys which need to be hold down for shortcuts

FUNCTIONKEYS
Possibility to activate buttons with function keys

COOLDRAGGING
Drag requesters around by clicking anywhere in window

REQUESTER LAYOUT

OUTERSPACINGX
 Horiz. space between win border and requester box

OUTERSPACINGY
 Vert. space between win border and requester box

INNERSPACINGX
 Min. horiz. space between requester box frame and text

INNERSPACINGY
 Min. vert. space between requester box frame and text

BODYGADGETSPACING
 Space between requester box and the buttons

BUTTONSPACING
 Minimum space between the buttons

BODYLOGOSPACING
 Minimum space between logo and requester text

LOGOALIGN
 Vertical alignment of logo

BODYXALIGN
 Horizontal alignment of requester text

BODYYALIGN
 Vertical alignment of requester text

REQTEXTALIGN
 Alignment of requester text lines

BUTTONALIGN
 Horizontal alignment of buttons

SAMEBUTTONWIDTH
 All buttons shall have the same width

REQPOS
 Position at which requester pops up

1.8 Configuration - General Settings

```
+-----+
| IMAGECACHES (NUMERIC)                               (default = 5) |
+-----+
```

Number of (external) images to cache in RAM so that they don't need to be reloaded each time. ReqAttack does not cache Bitmaps but files!

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

With this you can tell ReqAttack in which format the internal button image and requester logo bitmaps shall be created, which is especially interesting for GFX card owners. Possible VALUES are a combination of:

```
I: BMF_INTERLEAVED (create interleaved bitmaps whenever possible)
M: BMF_MINPLANES (use same color depth as screen)
F: FRIEND (bitmap shall have same format as screen bitmap)
```

Usually ReqAttack avoids creating an interleaved bitmap for images which have a mask (button images) if the screen is a standard Amiga screen. This is done because of the mask for interleaved bitmaps requiring much more memory. Interleaved bitmaps do have the advantage that there are less/hardly blink effects during blitting, but since the button images are rather small and not animated one usually doesn't notice it. Therefore by default 'I' is switched off. And for GFX card screens it has no meaning at all anyway.

'M' only makes sense in combination with 'F'. 'F' forces the creation of a bitmap in chunky format (if the screen is a GFX card screen). If this flag is not used, than the bitmap will always be in planar format and in CHIP RAM, which can cause blink effects. If additionally to the flag 'F' you also specify 'M', then also the color depth of the screen will be used instead of the maximum of 256 colors which a ReqAttack Image can have. So on 15/16 (hicolor) and 24 (truecolor) bit screens color remapping will not be necessary anymore which leads to better quality but also to increased memory usage.

In ReqAttack versions <= 1.0 none of this flags were used (BITMAPFLAGS = ""), that is, the bitmaps were always in planar non interleaved format.

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

Certain requesters can cause problems if ReqAttack is running. For example if a OS function call by ReqAttack for some reason causes the creation of a requester (maybe a filesystem process which does not look at pr_WindowPtr) then a deadlock situation blocking both ReqAttack and the task that wants to create the requester can arise. No task will be able to display requesters anymore (if one tries to do so, it will be blocked as well) and terminating ReqAttack will not be possible in such a case, either. There'll be no other way out than rebooting the computer. To avoid such problems you can tell ReqAttack to not patch certain requesters, that is, let the original OS requester routines handle everything:

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


1.9 Configuration - Requester look

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

Default logo to use for requesters with only one button.

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

Default logo to use for requesters with two buttons.

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

Default logo to use for requesters with three or more buttons.

```
+-----+
| LOGO (STRING)      (default = "" = use INFOLOGO, ASKLOGO or MULTIASKLOGO) |
+-----+
```

Logo to use for requester. Overrides INFOLOGO, ASKLOGO, MULTIASKLOGO.

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

Don't display any requester logo.

```
+-----+
| BACKGROUND (STRING)        (default = "" = standard background pattern) |
+-----+
```

Image to use as window background pattern.

```
+-----+
| BGPATTERN1PEN (COLOR)                                (default = SHINEPEN) |
+-----+
```

If you don't have an image as window background pattern, the standard Amiga requester background pattern will be used. This is a 2 color checkered pattern which by default uses SHINEPEN (white) as first color and BACKGROUNDPEN (grey) as second color. With BGPATTERN1PEN you can change the first color.

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

```
| BGPATTERN2PEN (COLOR)                                (default = BACKGROUNDPEN) |
+-----+-----+
```

If you don't have an image as window background pattern, the standard Amiga requester background pattern will be used. This is a 2 color checkered pattern which by default uses SHINEPEN (white) as first color and BACKGROUNDPEN (grey) as second color. With BGPATTERN2PEN you can change the second color.

```
+-----+-----+
| TEXTBOXBGPEN (COLOR)                                (default = BACKGROUNDPEN) |
+-----+-----+
```

Background color of the box containing the requester text and the logo.

```
+-----+-----+
| TEXTBOXTEXTPEN (COLOR)                              (default = TEXTPEN) |
+-----+-----+
```

Color to use for the requester text.

```
+-----+-----+
| OTEXTBOXEDGE1PEN (COLOR)                            (default = SHADOWPEN) |
+-----+-----+
```

Color of the left and top outer edges of the box containing the requester text and the logo.

```
+-----+-----+
| OTEXTBOXEDGE2PEN (COLOR)                            (default = SHINEPEN) |
+-----+-----+
```

Color of the right and bottom outer edges of the box containing the requester text and the logo.

```
+-----+-----+
| ITEXTBOXEDGE1PEN (COLOR)                            (default = <ignore>) |
+-----+-----+
```

Color of the left and top inner edges of the box containing the requester text and the logo. If you don't touch this, the inner left and top edges will not be drawn and therefore be only 1 pixel thick.

```
+-----+-----+
| ITEXTBOXEDGE2PEN (COLOR)                            (default = <ignore>) |
+-----+-----+
```

Color of the right and bottom inner edges of the box containing the requester text and the logo. If you don't touch this, the inner right and bottom edges will not be drawn and therefore be only 1 pixel thick.

1.10 Configuration - Button look

```
+-----+
| BUTTONSTYLE (...)                                (default = <ignore>) |
+-----+
```

Sets some of the following configuration items to predefined values depending on the VALUE you specify here. Possible VALUES are:

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

Try them out to see how it looks. Note that you can still override things set by BUTTONSTYLE by using the other button look configuration items somewhere later in the configuration script.

```
+-----+
| OBUTTONEDGE1PEN (COLOR)                          (default = SHINEPEN) |
+-----+
```

Color of the outer left and top button edges.

```
+-----+
| OBUTTONEDGE2PEN (COLOR)                          (default = SHADOWPEN) |
+-----+
```

Color of the outer right and bottom button edges.

```
+-----+
| IBUTTONEDGE1PEN (COLOR)                          (default = <ignore>) |
+-----+
```

Color of the inner left and top button edges. If you don't touch this, the inner left and top edges will not be drawn and therefore be only 1 pixel thick.

```
+-----+
| IBUTTONEDGE2PEN (COLOR)                          (default = <ignore>) |
+-----+
```

Color of the inner right and bottom button edges. If you don't touch this, the inner right and bottom edges will not be drawn and therefore be only 1 pixel thick.

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

```
| OBUTTONEDGE1SELPEN (COLOR) (default = SHADOWPEN) |
+-----+
```

Color of the outer left and top button edges when the button is selected (when you click it).

```
+-----+
| OBUTTONEDGE2SELPEN (COLOR) (default = SHINEPEN) |
+-----+
```

Color of the outer right and bottom button edges when the button is selected (when you click it).

```
+-----+
| IBUTTONEDGE1SELPEN (COLOR) (default = <ignore>) |
+-----+
```

Color of the inner left and top button edges when the button is selected (when you click it). If you don't touch this, the inner left and top edges will not be drawn and therefore be only 1 pixel thick.

```
+-----+
| IBUTTONEDGE2SELPEN (COLOR) (default = <ignore>) |
+-----+
```

Color of the inner right and bottom button edges when the button is selected (when you click it). If you don't touch this, the inner right and bottom edges will not be drawn and therefore be only 1 pixel thick.

```
+-----+
| OBUTTONEDGE1HIPEN (COLOR) (default = SHINEPEN) |
+-----+
```

Color of the outer left and top button edges when the button is highlighted.

```
+-----+
| OBUTTONEDGE2HIPEN (COLOR) (default = SHADOWPEN) |
+-----+
```

Color of the outer right and bottom button edges when the button is highlighted.

```
+-----+
| IBUTTONEDGE1HIPEN (COLOR) (default = <ignore>) |
+-----+
```

Color of the inner left and top button edges when the button is highlighted. If you don't touch this, the inner left and top edges will not be drawn and therefore be only 1 pixel thick.

Style of the button text. VALUE may be a combination of the following chars:

B: BOLD
I: ITALIC
U: UNDERLINED

```
+-----+
| BUTTONTEXTSELSTYLE (...)                               (default = <normal>) |
+-----+
```

Style of the button text when the button is selected (when you click it). VALUE may be a combination of the following chars:

B: BOLD
I: ITALIC
U: UNDERLINED

```
+-----+
| BUTTONTEXTSTYLE (...)                                 (default = <normal>) |
+-----+
```

Style of the button text when the button is highlighted. VALUE may be a combination of the following chars:

B: BOLD
I: ITALIC
U: UNDERLINED

```
+-----+
| SHIFTBUTTONTEXT (BOOL)                               (default = NO) |
+-----+
```

Shift button text and image one pixel to the right and to the bottom when button is selected (when you click it).

```
+-----+
| BUTTONISPACINGX (NUMERIC)                             (default = 4) |
+-----+
```

Space between the left/right button frame and the button text/image.

```
+-----+
| BUTTONISPACINGY (NUMERIC)                             (default = 2) |
+-----+
```

Minimum space between the upper/lower button frame and the button text.

```
+-----+
| BUTTONISPACINGY2 (NUMERIC)                             (default = 2) |
+-----+
```

-----+

Minimum space between the upper/lower button frame and the button image.

-----+

| BUTTONEXTRAWIDTH (NUMERIC) (default = 8) |

-----+

The minimum width of a button is calculated like this: width of the text + 2 x BUTTONISPACINGX + BUTTONEXTRAWIDTH. When button images are used, then width of the image + BIMAGETEXTSPACING is also added.

-----+

| BIMAGETEXTSPACING (NUMERIC) (default = 4) |

-----+

Minimum space between button image and button text.

-----+

| HILIGHTPEN (COLOR) (default = SHADOWPEN) |

-----+

Color of button hilight rectangle.

-----+

| HITHICKX (NUMERIC) (default = 1) |

-----+

Horizontal thickness in pixels of button highlight rectangle.

-----+

| HITHICKY (NUMERIC) (default = 1) |

-----+

Vertical thickness in pixels of button highlight rectangle.

-----+

| OKIMAGE (STRING) (default = \$OKIMAGE) |

-----+

Image to use for the leftmost (OK) button.

-----+

| CANCELIMAGE (STRING) (default = \$CANCELIMAGE) |

-----+

Image to use for the rightmost (CANCEL) button.

-----+


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

If you want to activate the buttons by using the functions keys (F1, F2, F3, ...) set this configuration item to YES.

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

If activated you can drag the requester window around by clicking anywhere in the window not just on the titlebar.

1.12 Configuration - Requester layout

```
+-----+-----+
| OUTERSPACINGX (NUMERIC) (default = 4) |
+-----+-----+
```

Space in pixels between the window border and the border of the requester bevel box (which contains the requester text and the logo) at the left and at the right side.

```
+-----+-----+
| OUTERSPACINGY (NUMERIC) (default = 4) |
+-----+-----+
```

Space in pixels between the upper window border and the upper border of the requester bevel box and between the lower window border and the buttons.

```
+-----+-----+
| INNERSPACINGX (NUMERIC) (default = 7) |
+-----+-----+
```

Minimum space in pixels between the border of the requester bevel box and the requester text/logo at the left and at the right side.

```
+-----+-----+
| INNERSPACINGY (NUMERIC) (default = 7) |
+-----+-----+
```

Minimum space in pixels between the border of the requester bevel box and the requester text/logo at the top and at the bottom side.

```
+-----+-----+
| BODYGADGETSPACING (NUMERIC) (default = 4) |
+-----+-----+
```

Space in pixels between the requester bevel box and the buttons.

Alignment of the requester text lines. Possible VALUES are:

```
LEFT
CENTER
RIGHT
```

Do not confuse REQTEXTALIGN with BODYXALIGN. REQTEXTALIGN determines the alignment of the single text lines within the imaginary box surrounding the whole requester text. With CENTER for example you get requesters which look like this:

```
+-----+
| Test Requester          |o|
+-----+
|
| +-----+   This is the   |
| |         |   text of a   |
| | LOGO |   test requester! |
| |         |
| +-----+
|
| +-----+   +-----+   |
| | OK |           ! Cancel | |
| +-----+   +-----+   |
+-----+
```

```
+-----+
| BUTTONALIGN (..)                               (default = SPREAD) |
+-----+
```

Horizontal alignment of the row of buttons when there is more space than necessary, that is when the requester bevel box is wider than the normal (minimum) width of the row of buttons. Possible VALUES are:

```
LEFT : |
+-----+
| +-----+ +-----+ +-----+ |
| | One | | Two | | Three | |
| +-----+ +-----+ +-----+ |
+-----+
```

```
CENTER: |
+-----+
| +-----+ +-----+ +-----+ |
| | One | | Two | | Three | |
| +-----+ +-----+ +-----+ |
+-----+
```

```
RIGHT : |
+-----+
| +-----+ +-----+ +-----+ |
| | One | | Two | | Three | |
| +-----+ +-----+ +-----+ |
+-----+
```

```

SPREAD: |
        +-----+
        | +-----+   +-----+   +-----+ |
        | | One |   | Two |   | Three | |
        | +-----+   +-----+   +-----+ |
        +-----+

```

```

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

```

All buttons shall have the same width. Usually the width of each button depends only on the width of it's text and it's image.

```

+-----+
| REQPOS (..)                                             (default = NORMAL) |
+-----+

```

Position at which requester window pops up. Possible VALUES are:

```

NORMAL: Upper left corner of screen.
CENTER: Center of screen.
MOUSE : Centered under mouse pointer.

```

1.13 Configuration - Check items

Following is a list of the available CHECKITEMs. The type of value expected by the CHECKITEM is shown between brackets. For strings you may specify an AmigaDOS pattern, for example "#?delete#?" to match any string "returned" by CHECKITEM that contains "delete" (like "delete", "I delete", "delete me" or "I want to delete this file").

```

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

```

The Task name of the program which opens the requester.

```

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

```

The Process name of the program which opens the requester. Sub-Tasks opened by programs often are not Processes (a Process is an extended Task - all programs started from Workbench or Shell are Processes - each Process is a Task, but not each Task is a Process) and therefore do not have a Process name.

```

+-----+

```

```
| PROGRAM (STRING) |
```

The Task name of the program which opens the requester or the Process name if there is no Task name.

```
| PROGRAM2 (STRING) |
```

The Process name of the program which opens the requester or the Task name if the Task is not a Process or if the Process does not have a name.

```
| SCREENTITLE (STRING) |
```

The title of the screen on which the requester pops up.

```
| REQTITLE (STRING) |
```

The window title of the requester.

```
| REQTEXT (STRING) |
```

The requester text. Lines are separated by linefeed characters (ASCII Code 10).

```
| SCREENDEPTH (NUMERIC) |
```

The depth (bits per color) of the screen on which the requester pops up. 1 means 2 colors, 4 means 16 colors, 8 means 256 colors, 16 means 65536 colors, etc. ($2^{\text{DEPTH}} = \text{NUMCOLORS}$).

```
| SCREENWIDTH (NUMERIC) |
```

Width of screen in pixels.

```
| SCREENHEIGHT (NUMERIC) |
```

Height of screen in pixels.

```
+-----+
| VISIBLEWIDTH (NUMERIC) |
+-----+
```

Visible width of screen in pixels.

```
+-----+
| VISIBLEHEIGHT (NUMERIC) |
+-----+
```

Visible height of screen in pixels.

```
+-----+
| RESOLUTIONX (NUMERIC) |
+-----+
```

DrawInfo resolution X value of the screen. The lower the value the higher the resolution.

```
+-----+
| RESOLUTIONY (NUMERIC) |
+-----+
```

DrawInfo resolution Y value of the screen. The lower the value the higher the resolution.

```
+-----+
| ASPECT (NUMERIC) |
+-----+
```

Combination of RESOLUTIONX and RESOLUTIONY. Returns 1 for 1:1 screens (320 x 256, 640 x 512, 1280 x 1024), 2 for 2:1 screens (640 x 256, 1280 x 512), 4 for 4:1 screens (1280 x 256), -2 for 1:2 screens (320 x 512, 640 x 1024), -4 for 1:4 screens (320 x 1024).

```
+-----+
| NUMBUTTONS (NUMERIC) |
+-----+
```

Number of buttons of the requester.

1.14 Datatypes

The ReqAttack Datatypes are mainly intended for image previewing. Unfortunately there are some problems with the ReqAttack AnimBrush Datatype. It is

buggy. Timing does not seem to work correctly - animation playback does not seem to start at first frame :(Any help would be appreciated!

It is possible to disable the ReqAttack AnimBrush Datatype on the fly by holding down LEFT ALT + LEFT SHIFT + CONTROL while the image is loaded. The ReqAttack Brush Datatype will take over and you will get a picture with all frames of the AnimBrush grouped together in one row. This will not work from Shell because of the special behaviour of Shells if you hold down the qualifiers (keys) mentioned above while pressing RETURN.

1.15 Bugs

- o Image loading is slow, at least on my 68030/50 A1200 :(
- o ReqAttack AnimBrush Datatype does not work correctly :((
- o ReqAttack might be dangerous. Try first with normal non-dangerous requesters to see if they work correctly (OK = okay and CANCEL = cancel). Be especially careful with "Format disk" requesters ;-)
- o If you get crashes after replying certain requesters make sure to use CyReq and/or StackAttack.
- o Make bug reports to georg.steger@rolmail.net!

1.16 Trouble Shooting

o The AssignWedge/WBAbout option of my XYZ commodity does not work ↔
anymore
if ReqAttack is running.

Use CyReq !

- o The computer crashes after replying "Please insert volume XYZ" requesters.

Use CyReq !

This seems to be a Stack problem of the ErrorReport() function from the dos.library. For more infos see history of

V 1.0
!

1.17 History

V 1.1
V 1.0
V 0.9
V 1.1 (07.09.1999) o New version of CyReq. Stack swapping is now ↔
also used
in the AutoRequest() patch.

- o The EasyRequestArgs() and AutoRequest() patches of ReqAttack, which are only activated if CyReq is not running, now are almost 100 % identical with the ones in CyReq, incl. Stack swapping etc.
 - o New configuration item
REQTEXTALIGN
which allows to
change the alignment of the requester text lines.
 - o New configuration item
FUNCTIONKEYS
which allows to
switch on button activation by function keys.
 - o Improved GFX card support thanks to Stephan Rupprecht. On hicolor and truecolor screens button images and requester logos are now by default created in the same color depth. By doing so color remapping is not required anymore and this leads to better quality. On 8 Bit GFX cards screens the bitmaps are now by default created in chunky format to avoid blink effects. Configuration item
BITMAPFLAGS
 - .
 - o New
compare operation
IN.
 - o New
CHECKITEM
 - PROGNAME2
 - . The description of PROG-
NAME was wrong, beside that.
 - o Animated logos are displayed with much less flickering on standard Amiga screens as well, if the screen is in interleaved format.
 - o The requesters were opened with IDCMP_CLOSEWINDOW despite the requester windows not having a close gadget. Because of this pressing ESCAPE did not work if one used the commodity NED.
 - o New configuration item
DONOTPATCH
 - .
 - o New tool WinInfo which is helpful to find out what task/process has opened a certain requester (or other window).
 - o It is now possible to stop the execution of the configuration file at any time by using
-

STOP

- V 1.0 (03.08.1999)
- o Lots of new configuration items to enhance the look of requesters (especially it's buttons) even more.
 - o For configuration items which expect a color you can now also specify a direct colormap (palette) pen by putting a '#' char in front of the number.
 - o Background pattern did not work at all on OS 3.0 and still had a few bugs even on OS 3.1.
 - o Transparency bug with CyberGraphX + SuperLayers should be fixed. Smart refresh windows created with LAYERS-NOBACKFILL are backfilled to color 0 with the standard Amiga layers library anyway, although they shouldn't. SuperLayers obviously does it correctly. I found this out only after someone reported the bug - I don't have a GFX card.
 - o The checkitem SCREENDDEPTH didn't return the correct value for HiColor and TrueColor screens. I used to look at DrawInfo->dri_Depth and this turned out to be never greater than 8. Strange ...
 - o Commodities Interface. You can disable ReqAttack temporarily (new opened requester will not be enhanced by Requester) and quit it with Exchange.
 - o New version of CyReq. On some systems the dos.library function ErrorReport() (which creates a new task to display the requester) seems to run out of stack very soon, so on some systems it was enough to enter something like "stupidtest:" in the Workbench's Execute Command window to make the computer crash once the requester asking for the disk "stupidtest:" was replied by the user. Now this should be fixed by using Stack swapping in CyReq's EasyRequestArgs() patch. The very strange thing is that ReqAttack since ever has used Stack swapping in BuildEasyRequestArgs() (which is called by CyReq's EasyRequestArgs() patch). On a PC running UAE + Picasso96 even CyReq running alone has caused the same crashes because of stack problems (I think). This is very strange since CyReq used very little stack and after all does not do nearly anything.
- I found the place in the ROM, where ErrorReport() creates it's task with too little stack so I might do a little utility to patch a ROM file and/or a BlizKick module sometime.
- o New Tool "QuitReqAttack". Can be used to quit ReqAttack from Shell scripts and ToolManager like programs.
-

o Improved docs a little bit.

V 0.9 (09.07.1999) o First public release

1.18 Future

- o Sound support (help wanted!)
- o Don't know. If you have some good ideas let me know!

1.19 Author

ReqAttack was created by:

Georg Steger
Hochlercher 30

I-39030 St. Johann/Ahrntal

E-Mail: georg.steger@rolmail.net

If you want to contact me and expect an answer send me an EMail! I will only respond by EMail. I hate snail mail ... My native language is German but you can also write in Italian or English.

***** ReqAttack is Shareware!! Read
Legal Stuff
! *****

1.20 Credits

- o Many thanks to Stephan Rupprecht for his RTG tips and routines.
 - o The incredible cool animated requester logo example images were done by Miikka Lehto (except the not so nice "Warn" which was done by me). Miikka has created different versions (size and number of colors) of each logo but to make sure that the ReqAttack archive does not get too big I have included only one version. You will find the remaining versions in a separate archive on Aminet!
 - o Some of the example background patterns were taken from the MUI archive. Unfortunately I can't remember where the others came from. If someone recognizes some of the images to be his work and wants to be credited in the docs or doesn't want the images to be included in the ReqAttack archive
-

then please let me know!

- o The standard button images were taken from MaxonDevelop 4.0 by Tilo Kühn.
- o The standard one-button requester logo was taken from a well known but terribly uncool operating system.

1.21 Legal Stuff

ReqAttack is Shareware. You can do some days or weeks of testings but if you keep using it you have to send me some money. Everything from 2000 Lire (2 DM, 1 US\$) upto 5000 Lire (5 DM, 3 US\$) is ok. If possible please do not send me the money in any other currency. I might eventually accept also a gift instead of the money, but ask me first (EMail)!!! Send everything to:

Georg Steger
Hochlercher 30

I-39030 St. Johann/Ahrntal - ITALY

EMail: georg.steger@rolmail.net

Important:

- o I cannot be held responsible for any damage that might occur when using ReqAttack!!!
- o I will not send out programs or program updates by snail mail!! Sorry, but look again at the extremely low registration fee - you simply cannot expect this from me.
- o Any questions? Send me an EMail! I will only respond by EMail. Sending letters by snail mail is too much work!

1.22 Final Notes

Have fun ;-)
