

**in**

<b>COLLABORATORS</b>
----------------------

	TITLE :  in		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 31, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>in</b>	<b>1</b>
1.1	Catharsis - Alpha release . . . . .	1
1.2	contents . . . . .	4

# Chapter 1

## in

### 1.1 Catharsis - Alpha release

Before you scream and move this complete package into Nirvana I want to tell you that this package was made in a hurry of two days (inclusive programming). So my excuses first.

Files comming with this package

Catharsis is a standard AmigaOS shared library designed to take a monochrome font on which a second one is created - which isn't a copy of the first one - where pixels will be only set if those pixels can make the font looking clearer. This works in this version only with tiny fonts up to 15 pixels in height. Additional pixels for larger fonts will only get set when in the source font this pixel isn't set (no modification). Since Catharsis will not patch anything, it creates this font at run-time - which takes a while.

Using Catharsis created fonts is a bit unusual and slows down the graphic-text-output by around 50 percent since you have to render the text twice; first using the standard text-font and then using the new created directly over the top of the first rendered where the second one is rendered in a different colour.

Example:

```
'RastPort' drawing handle
'TextFont' opened via Diskfont library
'CatharsisFont' created using Catharsis library
'TextPen' standard pen for text-output
'CatharsisPen' colour chosen by Catharsis library but obtained via Graphics library
```

```
SetAPen( RastPort, TextPen);
SetDrMd( RastPort, JAM2); we want to erase existing background
```

```

SetFont( RastPort, TextFont);
Move( RastPort, 50 , 20);
Text( RastPort, "Hello World", 11); text printed to display
! now we have to render the Catharsis character data over the top of the already ←
  rendered text !
SetAPen( RastPort, CartharsisPen);
SetDrMd( RastPort, JAM1); we want to overwrite
SetFont( RastPort, CartharsisFont);
Move( RastPort, 50 , 20);
Text( RastPort, "Hello World", 11); text printed to display (should look nicer)

```

A hint:

If you are using only one font in your display (window, layer, ←  
screen) and the  
background and foreground colour needn't to be change for different text- ←  
outputs,  
you can create a one-to-one copy of the original RastPort where the ←  
Catharsis  
related things can be performed, this saves a lot overhead and doesn' ←  
t lead the  
system to recalculate (minterns and other stuff, too) some things.

Above can then look like this:

```

'RastPort'  standard RastPort
'RastPort2' Catharsis's RastPort

```

```

Move( RastPort, 50 , 20);
Text( RastPort, "Hello World", 11);
Move( RastPort2, 50 , 20);
Text( RastPort2, "Hello World", 11);

```

you set the font and the drawing modes inclusive pens once for each ←  
RastPort, and  
afterwards you can use the RastPorts without changing fonts or pens nor ←  
drawing  
modes!

Function reference (only those of interest):

OpenCatharsisFontA

```

-----
struct TextFont *OpenCatharsisFontA( struct TextFont *font)
    D0:32      A0:32

```

This function will take an existing text-font and basing on it a fitting ←  
counterpart will be created -  
or an existing counterpart is returned if already by an other application created.

INPUTS:

TextFont standard TextFont returned by Graphics/OpenFont or Diskfont/ ←  
OpenDiskFont

RESULTS:

TextFont new created counterpart, viewed as standard TextFont or zero if

something went wrong

#### BUGS:

Does only create 'real' Catharsis fonts when the supplied TextFont is less than or equal to 15 pixels in height.  
 Does only allow at most 256 created fonts in memory.  
 Has yet no protection against modification while scanning the internal font-list .

#### CloseCatharsisFontA

```
void CloseCatharsisFontA( struct TextFont *font)
    A0:32
```

Closes the 'TextFont' returned by function OpenCatharsisFont(). If there is no more 'accessor' all resources of this 'TextFont' will be given back to the system.

#### INPUTS:

TextFont created counterpart by function OpenCatharsisFont()

#### TODO:

Should hold the font in memory as long no 'LibraryClose' command occurs, e.g. raised by '»avail flush«'.

#### CompressColorValueA

```
unsigned long CompressColorValueA( unsigned long red, unsigned long green,
    unsigned long blue)
    D0:32      D0:32  D1:32  D2:32
```

Converts RGB32 colour-component values to a standard 24bit colour value.

#### INPUTS:

red, green, blue each being a 32bit component

#### RESULTS:

aaRRGGBB value from the 32bits counterparts

#### CompatibleColorValueA

```
unsigned long CompatibleColorValueA( unsigned long value)
    D0:32      D0:8  !!!
```

Converts the colour-component value (8bit) into the RGB32 required format.

## INPUTS:

value 8bit component, either red, green or blue component per time

## RESULTS:

RGB32 32bit colour-component value

-----  
CatharsisColorValueA  
-----

```
unsigned long CatharsisColorValueA( unsigned long background, unsigned long ←
    foreground,
    d0:32      D0:32      D1:32
        unsigned long *red, unsigned long *green, unsigned long *blue)
        A0:32   A1:32   A2:32
```

Determine which colour suits best best when rendering the Catharsis character ←  
data over the  
originals.

INPUTS: background background colour-value, format aaRRGGBB  
foreground foreground colour-value, format aaRRGGBB  
red, gree, blue address of your variables where to store the RGB32 colour  
components, passing zero for each means not required

## RESULTS:

colour-value format aaRRGGBB  
\*red if non-zero passed, result of red value - RGB32 component  
\*green if non-zero passed, result of greenvalue - RGB32 component  
\*blue if non-zero passed, result of blue value - RGB32 component

## TODO:

Has to be extended or revised since it should give better results.

## 1.2 contents

## Package contents:

```
-----
pragma (dir)
    catharsis_lib.h
clib (dir)
    catharsis_protos.h
inline (dir)
    catharsis.h
proto (dir)
    catharsis.h
e (dir)
    catharsis.m                E-module
include (dir)                 NOTE: Upon install a subdir in YOUR "include"- ←
    directory labelled "catharsis" must be created for these files!
    catharsis.h
    catharsis.i                Asm
    catharsisbase.h
    catharsisbase.i           Asm
    catharsis_lib.i           Asm
```

---

example (dir)	Containing one, simple example
catharsis.library	
Example	
Example.c	
Example.project	MaxonC++ project file
Test	Test script with font "times/15"
libs (dir)	
catharsis.library	standard, AmigaOS shared library
Catharsis.project	MaxonC++ project file
LIB_Version.s	Assemble/compile and link in this order
LIB_Resident.s	
Catharsis_lib.c	
Catharsis.c	Result: catharsis.library
LIB_Resident.o	Already assembled
LIB_Version.o	
Catharsis.doc	Documentation
catharsis.fد	function description file
SnapShot.lbm	Snapshot of my Workbench with several fonts in ↵
conjunction with "catharsis.library"	

---