BITMAPLIB

Conversion program

COLLABORATORS			
	TITLE : BITMAPLIB		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Conversion program	October 9, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 BITMAPLIB **1**

1.1 Overview of BITMAPLIB 1

1.2 BITMAPLIB 1

1.3 BITMAPLIB 2

1.4 BITMAPLIB 2

1.5 BITMAPLIB 3

1.6 BITMAPLIB 3

1.7 BITMAPLIB 3

1.8 BITMAPLIB 3

Chapter 1

BITMAPLIB

1.1 Overview of BITMAPLIB

Overview

An Acid Software Library

Converted to AmigaGuide by

Red When Excited Ltd

Used with the permission of Acid Software

Edited, fixed and cleaned by Toby Zuijdveld 27/02/1999.
mailto:hotcakes@abacus.net.au

1.2 BITMAPLIB

```
Statement: BitMap
------------------------------------------------------------------------------
Modes  :
Syntax : BitMap BitMap#,Width,Height,Depth

BitMap creates and initializes a bitmap object. Once created, the
specified bitmap becomes the currently used bitmap. Width and Height
specify the size of the bitmap. Depth specifies how many colours may be
drawn onto the bitmap, and may be in the range one through six. The
actual colours available on a bitmap can be calculated using 2^depth.
For example, a bitmap of depth three allows for 2^3 or eight
colours.

3.0 BITMAP HANDLING

Blitz 2's Bitmap object has been upgraded to allow for interleaved
bitmaps:
```

```
NewType.Bitmap
_ebwidth[0]    ;00: for compatability.
_linemod.w    ;00: value to get from one scanline to next.
_height.w     ;02: currently pixel height - but open to commodore 'enhancement'.
_depth.w     ;04: number of bitplanes.
_pad.b[2]     ;06: nothing.
_data.l[8]    ;08: actual bitplane pointers.
_pad2.b[12]    ;40: zilch.
_flags.w     ;0=normal bitmap, <0=interleaved.
_bitplanemod.w  ;value to get from one bitplane to next. MAY BE 0!
_xclip.w     ;pixel width for render clipping
_yclip.w     ;pixel height for render clipping
_cclip.w     ;number of colours available on bitmap ( = 2^_depth)
_isreal.w     ;0=no bitmap here, <0=blitz created bitmap, >0=borrowed
         ;64: sizeof
End NEWTYPE
```

Also, many Blitz2 bitmap related commands have been altered to take this new object into account.

1.3 BITMAPLIB

```
Statement: LoadBitMap
------------------------------------------------------------------------------
Modes  :
Syntax : LoadBitMap BitMap#,Filename$[,Palette#]
```

LoadBitMap allows you to load an ILBM IFF graphic into a previously initialized bitmap object. You may optionally load in the graphics's colour palette into a palette object specified by Palette#. An error will be generated if the specified Filename$ is not in the correct IFF format.

1.4 BITMAPLIB

```
Statement: ScreensBitMap
------------------------------------------------------------------------------
Modes  :
Syntax : ScreensBitMap Screen#,BitMap#
```

Blitz 2 allows you the option of attaching a bitmap object to any Intuition Screens you open. If you open a Screen without attaching a bitmap, a bitmap will be created anyway. You may then find this bitmap using the ScreensBitMap command. Once ScreensBitMap is executed, the specified bitmap becomes the currently used bitmap.

1.5 BITMAPLIB

```
Statement: CopyBitMap
------------------------------------------------------------------------------
Modes  :
Syntax : CopyBitMap Src BitMap#,Dest BitMap#

CopyBitMap will make an exact copy of a bitmap object into another
bitmap object. The first BitMap# parameter specifies the source bitmap
for the copy, the second BitMap# the destination.

Any graphics rendered onto the source bitmap will also be
copied.
```

1.6 BITMAPLIB

```
Statement: ShapesBitMap
------------------------------------------------------------------------------
Modes  :
Syntax : ShapesBitMap Shape#,BitMap#

ShapesBitMap creates a dummy BitMap so drawing commands can be used
directly on a shapes image data.
```

1.7 BITMAPLIB

```
Statement: CludgeBitMap
------------------------------------------------------------------------------
Modes  :
Syntax : CludgeBitMap BitMap#,Width,Height,Depth,Memory

CludgeBitMap will create a bitmap object with the proportions for that
specified using the memory location given. Of course, the memory
location specified must be in chipmem and it is upto the user to ensure
that sufficient memory has been allocated. This commands is most useful
for games where memory fragmentation can be a big problem, by
allocating one block of memory on program initialisation for all
bitmaps CludgeBitMap can be used so that creating and freeing of
BitMaps is not necessary.
```

1.8 BITMAPLIB

```
                .-------------------------------------------------------------------------
|                                   BITMAPLIB                                   |
`-------------------------------------------------------------------------------'
```

Overview

Command Index

BitMap

CludgeBitMap

CopyBitMap

LoadBitMap

ScreensBitMap

ShapesBitMap