NAME

bitmap, bmtoa, atobm – bitmap editor and converter utilities for the X Window System

SYNOPSIS

bitmap [–*options* . . .] [*filename*] [*basename*]

bmtoa [**–chars** . . .] [*filename*]

atobm [**–chars** *cc*] [–**name** *variable*] [–**xhot** *number*] [–**yhot** *number*] [*filename*]

DESCRIPTION

The **bitmap** program is a rudimentary tool for creating or editing rectangular images made up of 1's and 0's. Bitmaps are used in X for defining clipping regions, cursor shapes, icon shapes, and tile and stipple patterns.

The **bmtoa** and **atobm** filters convert **bitmap** files (FILE FORMAT) to and from ASCII strings. They are most commonly used to quickly print out bitmaps and to generate versions for including in text.

COMMAND LINE OPTIONS

The **bitmap** command supports the standard X Toolkit command line arguments (see **X11**(7)). The following additional arguments are supported as well.

–**size** *WIDTHxHEIGHT*
Specifies size of the grid in squares.

–**sw** *dimension*
Specifies the width of squares in pixels.

–**sh** *dimension*
Specifies the height of squares in pixels.

–**gt** *dimension*
Grid tolerance. If the square dimensions fall below the specified value, grid will be automatically turned off.

–**grid, +grid**
Turns on or off the grid lines.

–**axes, +axes**
Turns on or off the major axes.

–**dashed, +dashed**
Turns on or off dashing for the frame and grid lines.

–**stippled, +stippled**
Turns on or off stippling of highlighted squares.

–**proportional, +proportional**
Turns proportional mode on or off. If proportional mode is on, square width is equal to square height. If proportional mode is off, **bitmap** will use the smaller square dimension, if they were initially different.

–**dashes** *filename*
Specifies the bitmap to be used as a stipple for dashing.

–**stipple** *filename*
Specifies the bitmap to be used as a stipple for highlighting.

–**hl** *color*
Specifies the color used for highlighting.

–**fr** *color*
Specifies the color used for the frame and grid lines.

filename
Specifies the bitmap to be initially loaded into the program. If the file does not exist, **bitmap** will assume it is a new file.

basename
Specifies the basename to be used in the C code output file. If it is different than the basename in the working file, **bitmap** will change it when saving the file.

The **bmtoa** command accepts the following option:

–**chars** *cc*
This option specifies the pair of characters to use in the string version of the bitmap. The first character is used for 0 bits and the second character is used for 1 bits. The default is to use dashes (–) for 0's and number signs (#) for 1's.

The **atobm** command accepts the following options:

–**chars** *cc*
This option specifies the pair of characters to use when converting string bitmaps into arrays of numbers. The first character represents a 0 bit and the second character represents a 1 bit. The default is to use dashes (–) for 0's and number signs (#) for 1's.

–**name** *variable*
This option specifies the variable name to be used when writing out the bitmap file. The default is to use the basename of the *filename* command line argument or leave it blank if the standard input is read.

–**xhot** *number*
This option specifies the X coordinate of the hotspot. Only positive values are allowed. By default, no hotspot information is included.

–**yhot** *number*
This option specifies the Y coordinate of the hotspot. Only positive values are allowed. By default, no hotspot information is included.

USAGE

The **bitmap** command displays grid in which each square represents a single bit in the picture being edited. Actual size of the bitmap image, as it would appear normaly and inverted, can be obtained by pressing **Meta-I** key. You are free to move the image popup out of the way to continue editing. Pressing the left mouse button in the popup window or **Meta-I** again will remove the real size bitmap image.

If the bitmap is to be used for defining a cursor, one of the squares in the images may be designated as the hot spot. This determines where the cursor is actually pointing. For cursors with sharp tips (such as arrows or fingers), this is usually at the end of the tip; for symmetric cursors (such as crosses or bullseyes), this is usually at the center.

Bitmaps are stored as small C code fragments suitable for including in applications. They provide an array of bits as well as symbolic constants giving the width, height, and hot spot (if specified) that may be used in creating cursors, icons, and tiles.

EDITING

To edit a bitmap image simply click on one of the buttons with drawing commands (**Point, Curve, Line, Rectangle,** etc.) and move the pointer into the bitmap grid window. Press one of the buttons on your mouse and the appropriate action will take place. You can either set, clear or invert the grid squares. Setting a grid square corresponds to setting a bit in the bitmap image to 1. Clearing a grid square corresponds to setting a bit in the bitmap image to 0. Inverting a grid square corresponds to changing a bit in the bitmap image from 0 to 1 or 1 to 0, depending what its previous state was. The default behavior of mouse buttons is as specified below.

MouseButton1	Set
MouseButton2	Invert
MouseButton3	Clear
MouseButton4	Clear
MouseButton5	Clear

This default behavior can be changed by setting the button function resources. An example is provided below.

```
bitmap*button1Function: Set
bitmap*button2Function: Clear
bitmap*button3Function: Invert
etc.
```

The button function applies to all drawing commands, including copying, moving and pasting, flood filling and setting the hot spot.

DRAWING COMMANDS

Here is the list of drawing commands accessible through the buttons at the left side of the application's window. Some commands can be aborted by pressing A inside the bitmap window, allowing the user to select different guiding points where applicable.

Clear
This command clears all bits in the bitmap image. The grid squares will be set to the background color. Pressing C inside the bitmap window has the same effect.

Set This command sets all bits in the bitmap image. The grid squares will be set to the foreground color. Pressing S inside the bitmap window has the same effect.

Invert
This command inverts all bits in the bitmap image. The grid squares will be inverted appropriately. Pressing I inside the bitmap window has the same effect.

Mark
This command is used to mark an area of the grid by dragging out a rectangular shape in the highlighting color. Once the area is marked, it can be operated on by a

number of commands (see **Up, Down, Left, Right, Rotate, Flip, Cut,** etc.) Only one marked area can be present at any time. If you attempt to mark another area, the old mark will vanish. The same effect can be achieved by pressing **Shift-MouseButton1** and dragging out a rectangle in the grid window. Pressing **Shift-MouseButton2** will mark the entire grid area.

Unmark
This command will cause the marked area to vanish. The same effect can be achieved by pressing **Shift-MouseButton3**.

Copy
This command is used to copy an area of the grid from one location to another. If there is no marked grid area displayed, **Copy** behaves just like **Mark** described above. Once there is a marked grid area displayed in the highlighting color, this command has two alternative behaviors. If you click a mouse button inside the marked area, you will be able to drag the rectangle that represents the marked area to the desired location. After you release the mouse button, the area will be copied. If you click outside the marked area, **Copy** will assume that you wish to mark a different region of the bitmap image, thus it will behave like **Mark** again.

Move
This command is used to move an area of the grid from one location to another. Its behavior resembles the behavior of **Copy** command, except that the marked area will be moved instead of copied.

Flip Horizontally
This command will flip the bitmap image with respect to the horizontal axes. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing F inside the bitmap window has the same effect.

Up This command moves the bitmap image one pixel up. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing UpArrow inside the bitmap window has the same effect.

Flip Vertically
This command will flip the bitmap image with respect to the vertical axes. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing V inside the bitmap window has the same effect.

Left
This command moves the bitmap image one pixel to the left. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing LeftArrow inside the bitmap window has the same effect.

Fold
This command will fold the bitmap image so that the opposite corners become adjacent. This is useful when creating bitmap images for tiling. Pressing F inside the bitmap window has the same effect.

Right
This command moves the bitmap image one pixel to the right. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing

RightArrow inside the bitmap window has the same effect.

Rotate Left

This command rotates the bitmap image 90 degrees to the left (counter clockwise.) If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing L inside the bitmap window has the same effect.

Down

This command moves the bitmap image one pixel down. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing DownArrow inside the bitmap window has the same effect.

Rotate Right

This command rotates the bitmap image 90 degrees to the right (clockwise.) If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing R inside the bitmap window has the same effect.

Point

This command will change the grid squares underneath the mouse pointer if a mouse button is being pressed down. If you drag the mouse button continuously, the line may not be continuous, depending on the speed of your system and frequency of mouse motion events.

Curve

This command will change the grid squares underneath the mouse pointer if a mouse button is being pressed down. If you drag the mouse button continuously, it will make sure that the line is continuous. If your system is slow or **bitmap** receives very few mouse motion events, it might behave quite strangely.

Line

This command will change the grid squares in a line between two squares. Once you press a mouse button in the grid window, **bitmap** will highlight the line from the square where the mouse button was initially pressed to the square where the mouse pointer is located. By releasing the mouse button you will cause the change to take effect, and the highlighted line will disappear.

Rectangle

This command will change the grid squares in a rectangle between two squares. Once you press a mouse button in the grid window, **bitmap** will highlight the rectangle from the square where the mouse button was initially pressed to the square where the mouse pointer is located. By releasing the mouse button you will cause the change to take effect, and the highlighted rectangle will disappear.

Filled Rectangle

This command is identical to **Rectangle**, except at the end the rectangle will be filled rather than outlined.

Circle

This command will change the grid squares in a circle between two squares. Once you press a mouse button in the grid window, **bitmap** will highlight the circle from the square where the mouse button was initially pressed to the square where the mouse pointer is located. By releasing the mouse button you will cause the change to take effect, and the highlighted circle will disappear.

Filled Circle
This command is identical to **Circle**, except at the end the circle will be filled rather than outlined.

Flood Fill
This command will flood fill the connected area underneath the mouse pointer when you click on the desired square. Diagonally adjacent squares are not considered to be connected.

Set Hot Spot
This command designates one square in the grid as the hot spot if this bitmap image is to be used for defining a cursor. Pressing a mouse button in the desired square will cause a diamond shape to be displayed.

Clear Hot Spot
This command removes any designated hot spot from the bitmap image.

Undo
This command will undo the last executed command. It has depth one, that is, pressing **Undo** after **Undo** will undo itself.

FILE MENU

The File menu commands can be accessed by pressing the File button and selecting the appropriate menu entry, or by pressing Ctrl key with another key. These commands deal with files and global bitmap parameters, such as size, basename, filename etc.

New
This command will clear the editing area and prompt for the name of the new file to be edited. It will not load in the new file.

Load
This command is used to load a new bitmap file into the bitmap editor. If the current image has not been saved, user will be asked whether to save or ignore the changes. The editor can edit only one file at a time. If you need interactive editing, run a number of editors and use cut and paste mechanism as described below.

Insert
This command is used to insert a bitmap file into the image being currently edited. After being prompted for the filename, click inside the grid window and drag the outlined rectangle to the location where you want to insert the new file.

Save
This command will save the bitmap image. It will not prompt for the filename unless it is said to be <none>. If you leave the filename undesignated or –, the output will be piped to stdout.

Save As
This command will save the bitmap image after prompting for a new filename. It should be used if you want to change the filename.

Resize
This command is used to resize the editing area to the new number of pixels. The

size should be entered in the WIDTHxHEIGHT format. The information in the image being edited will not be lost unless the new size is smaller that the current image size. The editor was not designed to edit huge files.

Rescale
This command is used to rescale the editing area to the new width and height. The size should be entered in the WIDTHxHEIGHT format. It will not do antialiasing and information will be lost if you rescale to the smaller sizes. Feel free to add you own algorithms for better rescaling.

Filename
This command is used to change the filename without changing the basename nor saving the file. If you specify – for a filename, the output will be piped to stdout.

Basename
This command is used to change the basename, if a different one from the specified filename is desired.

Quit
This command will terminate the bitmap application. If the file was not saved, user will be prompted and asked whether to save the image or not. This command is preferred over killing the process.

EDIT MENU

The Edit menu commands can be accessed by pressing the Edit button and selecting the appropriate menu entry, or by pressing Meta key with another key. These commands deal with editing facilities such as grid, axes, zooming, cut and paste, etc.

Image
This command will display the image being edited and its inverse in its actual size in a separate window. The window can be moved away to continue with editing. Pressing the left mouse button in the image window will cause it to disappear from the screen.

Grid
This command controls the grid in the editing area. If the grid spacing is below the value specified by gridTolerance resource (8 by default), the grid will be automatically turned off. It can be enforced by explicitly activating this command.

Dashed
This command controls the stipple for drawing the grid lines. The stipple specified by dashes resource can be turned on or off by activating this command.

Axes
This command controls the highlighting of the main axes of the image being edited. The actual lines are not part of the image. They are provided to aid user when constructing symmetrical images, or whenever having the main axes highlighted helps your editing.

Stippled
This command controls the stippling of the highlighted areas of the bitmap image. The stipple specified by stipple resource can be turned on or off by activating this command.

Proportional
This command controls the proportional mode. If the proportional mode is on, width and height of all image squares are forced to be equal, regardless of the proportions of the bitmap window.

Zoom
This command controls the zoom mode. If there is a marked area of the image already displayed, bitmap will automatically zoom into it. Otherwise, user will have to highlight an area to be edited in the zoom mode and bitmap will automatically switch into it. One can use all the editing commands and other utilities in the zoom mode. When you zoom out, undo command will undo the whole zoom session.

Cut This commands cuts the contents of the highlighted image area into the internal cut and paste buffer.

Copy
This command copies the contents of the highlighted image area into the internal cut and paste buffer.

Paste
This command will check if there are any other bitmap applications with a highlighted image area, or if there is something in the internal cut and paste buffer and copy it to the image. To place the copied image, click in the editing window and drag the outlined image to the position where you want to place i, and then release the button.

CUT AND PASTE

Bitmap supports two cut and paste mechanisms; the internal cut and paste and the global X selection cut and paste. The internal cut and paste is used when executing copy and move drawing commands and also cut and copy commands from the edit menu. The global X selection cut and paste is used whenever there is a highlighted area of a bitmap image displayed anywhere on the screen. To copy a part of image from another bitmap editor simply highlight the desired area by using the Mark command or pressing the shift key and dragging the area with the left mouse button. When the selected area becomes highlighted, any other applications (such as xterm, etc.) that use primary selection will discard their selection values and unhighlight the appropriate information. Now, use the Paste command for the Edit menu or control mouse button to copy the selected part of image into another (or the same) bitmap application. If you attempt to do this without a visible highlighted image area, the bitmap will fall back to the internal cut and paste buffer and paste whatever was there stored at the moment.

WIDGETS

Below is the widget structure of the **bitmap** application. Indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name. All widgets except the bitmap widget are from the standard Athena widget set.

```
Bitmap bitmap
    TransientShell image
        Box box
            Label normalImage
```

```
                Label invertedImage
TransientShell input
        Dialog dialog
                Command okay
                Command cancel
TransientShell error
        Dialog dialog
                Command abort
                Command retry
TransientShell qsave
        Dialog dialog
                Command yes
                Command no
                Command cancel
Paned parent
        Form formy
                MenuButton fileButton
                SimpleMenu fileMenu
                        SmeBSB  new
                        SmeBSB  load
                        SmeBSB  insert
                        SmeBSB  save
                        SmeBSB  saveAs
                        SmeBSB  resize
                        SmeBSB  rescale
                        SmeBSB  filename
                        SmeBSB  basename
                        SmeLine line
                        SmeBSB  quit
                MenuButton editButton
                SimpleMenu editMenu
                        SmeBSB  image
                        SmeBSB  grid
                        SmeBSB  dashed
                        SmeBSB  axes
                        SmeBSB  stippled
                        SmeBSB  proportional
                        SmeBSB  zoom
                        SmeLine line
                        SmeBSB  cut
                        SmeBSB  copy
                        SmeBSB  paste
                Label status
        Pane pane
                Bitmap bitmap
```

```
Form form
        Command clear
        Command set
        Command invert
        Toggle  mark
        Command unmark
        Toggle  copy
        Toggle  move
        Command flipHoriz
        Command up
        Command flipVert
        Command left
        Command fold
        Command right
        Command rotateLeft
        Command down
        Command rotateRight
        Toggle  point
        Toggle  curve
        Toggle  line
        Toggle  rectangle
        Toggle  filledRectangle
        Toggle  circle
        Toggle  filledCircle
        Toggle  floodFill
        Toggle  setHotSpot
        Command clearHotSpot
        Command undo
```

COLORS

If you would like bitmap to be viewable in color, include the following in the #ifdef COLOR section of the file you read with xrdb:

∗customization: –color

This will cause bitmap to pick up the colors in the app-defaults color customization file: /usr/openwin/lib/app-defaults/Bitmap.

BITMAP WIDGET

Bitmap widget is a stand-alone widget for editing raster images. It is not designed to edit large images, although it may be used in that purpose as well. It can be freely incorporated with other applications and used as a standard editing tool. The following are the resources provided by the bitmap widget.

Bitmap Widget

Header file	Bitmap.h
Class	bitmapWidgetClass
Class Name	Bitmap
Superclass	Bitmap

All the Simple Widget resources plus . . .

Name	**Class**	**Type**	**Default Value**
foreground	Foreground	Pixel	XtDefaultForeground
highlight	Highlight	Pixel	XtDefaultForeground
framing	Framing	Pixel	XtDefaultForeground
gridTolerance	GridTolerance	Dimension	8
size	Size	String	32x32
dashed	Dashed	Boolean	True
grid	Grid	Boolean	True
stippled	Stippled	Boolean	True
proportional	Proportional	Boolean	True
axes	Axes	Boolean	False
squareWidth	SquareWidth	Dimension	16
squareHeight	SquareHeight	Dimension	16
margin	Margin	Dimension	16
xHot	XHot	Position	NotSet (–1)
yHot	YHot	Position	NotSet (–1)
button1Function	Button1Function	DrawingFunction	Set
button2Function	Button2Function	DrawingFunction	Invert
button3Function	Button3Function	DrawingFunction	Clear
button4Function	Button4Function	DrawingFunction	Invert
button5Function	Button5Function	DrawingFunction	Invert
filename	Filename	String	None ("")
basename	Basename	String	None ("")

AUTHOR

Davor Matic, X Consortium