







# Making New Bulbs for MacLampsX

Last updated: December 2008

## How Bulbs Work

Each bulb in MacLampsX consists of a special kind of folder called a package. This just means that the folder has an extension and that the Finder displays allows users to manipulate it as though it is a single file. A bulb package for MacLampsX looks like this:

▼  spacer.bulb	Bulb package – MLX bulbs are identified by the extension <code>.bulb</code> .
▼  Contents	<code>Contents</code> – Everything useful goes in here. Don't forget this.
 bulb.plist	<code>bulb.plist</code> describes the bulb – name, size, etc.
 corners.png	<code>corners.png</code> – PNG image with graphics for the corners of the screen.
 display.png	<code>display.png</code> – optional image to use in bulb list.
 sides.png	<code>sides.png</code> contains the graphics used along the sides of the screen.

To access the files inside a bulb, control-click on the bulb in the Finder, and select "Show Package Contents..." from the contextual menu.

An example bulb has been included with this document to help you follow along as you read this document, and to serve as a starting point for you to create your own bulbs. In addition, the photoshop file for the light base used in some of the default bulbs for MLX included if you wish to use it for your bulb files.

## Creating Images

`display.png` is a single 32x32 image which can be displayed in the bulb list when the image that MacLampsX selects automatically is undesirable. MLX only uses `display.png` when the appropriate option is enabled in `bulb.plist`.

You can choose to provide graphics 0, 1, or 4 sides of the screen in your bulb (set in `bulb.plist`). 4 will allow you to have your bulb appear differently on each side of the screen. 1 will make your bulb appear the same no matter which side of the screen it is placed on. 0 will cause MacLampsX to prohibit users from placing your bulb on the sides of the screen at all (corners only).

Corresponding to the number of sides you choose to pick, you will have either 1 or 4 rows in `sides.png` (or you can omit the file altogether for 0). With four sides, the rows from top to bottom of the image are: 1) bottom of screen, 2) left side, 3) top, 4) right.

Each column of `sides.png` represents a flavor (color variations or whatever) of your bulb. Blocks of multiple columns together are states (on/off). For example, if your bulb has five colors and on/off states, the image would have 5 columns of different colors in the on state, followed by 5 in the off state. By default the first state is considered 'on' and the second is considered 'off'. Any additional states will be displayed only in chase mode, and in the order they come in the file. You can provide as many flavors/states as you want, within reason.

Finally, there is an option to have rectangular bulbs – for example if you wanted to have a 1x32 'line' bulb. However, the graphics must still be laid out on a square pixel grid in `sides.png` – for the 'line' bulb, each bulb still gets a 32x32 square, which it should live at the lower-left corner of. If your bulb provides graphics for 4 sides, the aspect ratio for the left and right sides is assumed to be swapped – a bulb that's 1x32 on the top and bottom is assumed to be 32x1 on the left and right.

`corners.png` works very much like `sides.png` except that the rows are bottom-right, bottom-left, top-left, top-right if you provide images for four corners, and that you cannot provide multiple flavors for the corners. In addition, corner bulbs must be square. Sorry.

## Editing bulb.plist

`bulb.plist` is a format called a XML Property List. The easiest way to edit these files is with software that's specifically designed to do so – for example, the free Property List Editor that's included with Apple's Developer tools – however you can also edit them by hand in a text editor if you prefer. See [this document](#) with more info on property lists.

As its name may suggest, `bulb.plist` contains a list of properties of your bulb. The table below describes each property that you can set. Some properties are valid for all bulbs, and some are valid for a particular version of the bulb format.

Property	Type	Description
bulbHumanName	String	The name displayed in the bulb list
bulbDescription	String	The description in the bulb list
bulbName	String	Uniquely identifies your bulb to MLX. Please use a reverse domain identifier here to avoid name conflicts. Something like <code>com.yourname.MacLampsXBulb.bulbname</code>
authorName	String	Displayed to the user in detailed bulb info
authorContact	String	Displayed to the user in detailed bulb info
designDate	String	Displayed to the user in detailed bulb info
variations	Integer	The number of variations (colors or whatever) your bulb contains. For example, the default Outdoor bulb has 5. You must have at least 1 variation.
lightsVersion	Integer	Version of the bulb format you're using. At present, this can be either 1 or 2. Version 2 bulbs can only be used with MLX 2.0 or later, but offer more features. V1 bulbs can be used with any version of MLX.

#### The following properties apply only to lightsVersion=2

width	Integer	Pixel width of bulbs. Only applies to sides. If you are providing 4 sides, this number will be the width for the images along the top and bottom, and the height for the images along the left and right.
height	Integer	Pixel height of bulbs. Works analogously to <code>width</code> .
cornerSize	Integer	Pixel width and height of corner graphics. Corners are square.
cornerCount	Integer	The number of corners your bulb includes. Must be 0, 1, or 4. 0 prevents the bulb from being used in corners, and allows you to omit <code>corners.png</code> . 1 indicates that the same graphic should be used regardless of corner being drawn. 4 indicates a unique graphic for each corner.
sideCount	Integer	The number of sides in your bulb. Works analogously to <code>cornerCount</code> .
states	Integer	The number of states. The first state is considered 'on', and the second is 'off'. Additional states have no special designations. You must have at least 1 state.
hasPreview	Boolean	<code>True</code> indicates that MLX should look for and use <code>display.png</code> as the preview image in the bulb list. <code>False</code> indicates that MLX should generate the preview on its own.

#### The following properties apply only to lightsVersion=1

flashes	Boolean	replaced by <code>states</code> in v2. <code>True</code> means 2 states, <code>False</code> means one state.
hasFourCorners	Boolean	replaced by <code>cornerCount</code> in v2. <code>True</code> means 4 corners, <code>False</code> means 1.
hasRotations	Boolean	replaced by <code>sidesCount</code> in v2. <code>True</code> means 4 sides, <code>False</code> means 1.

Of note, version 1 bulbs are considered to be 32px square.

## Common Problems

If you simply can't get MacLampsX to load your bulb at all,

1. You may want to check the console to see if MacLampsX is reporting any useful errors. It logs some problems to the console to help you out. The console can be found in `/Applications/Utilities/`.
2. Check that the package structure is right. People have been known to forget the `Contents` folder in the past.
3. If you've been editing the property list by hand, try it in an editor to make sure you didn't mess up the XML.
4. If things still don't work, you may've set one of the properties in a way that MLX can't figure out. Read over the list and try to isolate the property that's the issue.

If MLX displays your bulbs, but they are scaled or cropped (and you are sure you set the width and height right), you may have accidentally saved a resolution other than screen resolution (72dpi by default) into your files. To fix this, simply open your bulb in an image editor that can set resolution (like Preview in 10.5), and set your image to 72 dpi (make sure it doesn't resample the image when it does this), then save changes.

## Tips for Distributing Bulbs

I highly encourage you to share any bulbs you make with the world at large. You can easily do this on your website, but you can also email me your bulb, so that I can add it to the official bulb archive, which is where most people will end up looking for bulbs. Keep in mind that your bulb is a package, not a single file, so in order to distribute it, you need to pack

it up into a single file. This is most easily done by control-clicking on the bulb and selecting "Compress mybulb.bulb" from the contextual menu, but before you do that...

The remainder of this section focuses on little perfectionist things that I like to do before I distribute bulbs. Some of it assumes familiarity with the terminal, and it's all fine to skip, especially since, if you send me a bulb to add to the archive, I'll be doing most of these steps for you anyway.

1. The PNG images in your bulb are likely not optimally compressed, and it's nice to do so to save a little space in the downloads and just to get rid of extra fluff. Both Preview (as of 10.5) and Photoshop's Save For Web dialog do reasonably well, but the best results are almost always achieved by optipng or pngcrush, both of which are opensource utilities designed to make PNG files smaller. You'll need to figure out how to install one of these if you want to use it, though I'd recommend using either fink or macports to help you with the installation. I usually use pngcrush: `'pngcrush -rem alla -brute inputfile.png outputfile.png'`.
2. Mac OS creates files called `.DS_Store` in which it saves the Finder's view settings in each folder. I like to remove these. Close any Finder windows showing the contents of your bulb. Then, in the terminal, just `cd` to the bulb directory, then do `'rm .DS_Store'`, then `'cd Contents'`, and `'rm .DS_Store'` one more time.
3. Finally you should archive and compress your bulb. The above mentioned method is very easy, and works just fine, but I sometimes like to try to squeeze the size down a little more, using tar and gzip in the terminal: `'tar -c bulbpath.bulb | gzip -9 -c > outfile.tgz'`.