

ICONW

This is a package of functions for building small windows of arbitrary shape, principally for use as icons for shrinking windows; i.e., these functions are likely to be invoked from within the `ICONFN` of a window.

An icon is specified by supplying its image (a bit map) and a mask that specifies its shape. The mask is a bit map of the same dimensions as the image whose bits are on (black) in those positions considered to be in the image, off (white) in those positions where the background should "show through." By using the mask and appropriate window functions, the icon package maintains the illusion that the icon window is nonrectangular, even though the actual window itself is rectangular. The illusion is not complete, of course. For example, if you try to select what looks like the background (or an occluded window) around the icon but still within its rectangular perimeter, the icon window itself is selected. Also, if you move a window occluded by an icon, the icon never notices that the background changed behind it.

Icons created with this package can also have "titles"; some part of the image can be filled with text computed at the time the icon is created, or even changed after creation.

Creating Icons

(`ICONW IMAGE MASK POSITION NOOPENFLG`) [Function]

Creates a window at `POSITION`, or prompts for a position if `POSITION` is `NIL`. The window is borderless, and filled with `IMAGE`, as cookie-cut by `MASK`. If `MASK` is `NIL`, the image is considered rectangular (i.e., `MASK` defaults to a black bit map of the same dimensions as `IMAGE`). If `NOOPENFLG` is `T`, the window is returned unopened.

(`TITLEDICONW ICON TITLE FONT POSITION NOOPENFLG JUST BREAKCHARS OPERATION`) [Function]

Creates a titled icon at `POSITION`, or prompts for a position if `POSITION` is `NIL`. If `NOOPENFLG` is `T`, the window is returned unopened. The argument `ICON` is an instance of the record `TITLEDICON`, which specifies the icon image and mask, as with `ICONW`, and a region within the image to be used for displaying the title. Thus, the `ICON` argument is usually of the form

```
(create TITLEDICON ICON_ someIconImage
      MASK_ iconMask TITLEREG_ someRegionWithinICON).
```

The title region is specified in icon-relative coordinates, i.e., the lower-left corner of the image bit map is (0, 0). The mask can be `NIL` if the icon is rectangular. The image should be white where it is covered by the title region (in any event, `TITLEDICONW` clears the region before printing on it).

The title is printed into the specified region in the image, using *FONT*, which if *NIL* defaults to the value of *DEFAULTICONFONT*, initially Helvetica 10. The title is broken into multiple lines if necessary; *TILEDICONW* attempts to place the breaks at characters that are in the list of character codes *BREAKCHARS*. *BREAKCHARS* defaults to (CHARCODE (SPACE *ŷ*)). In addition, line breaks are forced by any carriage returns in *TITLE*, independent of *BREAKCHARS*. *BREAKCHARS* is ignored as needed if a long title would not otherwise fit in the specified region. For convenience, *BREAKCHARS* = *FILE* means the title is a file name, so break at file name field delimiters.

The argument *JUST* indicates how the text should be justified relative to the region—it is an atom or list of atoms chosen from *TOP*, *BOTTOM*, *LEFT*, or *RIGHT*, which indicate the vertical positioning (flush to top or bottom) and/or horizontal positioning (flush to left edge or right). Where not indicated, the text is centered.

The argument *OPERATION* is a display stream operation indicating how the title should be printed. If *OPERATION* is *INVERT*, then the title is printed white-on-black. The default *OPERATION* is *REPLACE*, meaning black-on-white. *ERASE* is the same as *INVERT*; *PAINT* is the same as *REPLACE*.

For convenience, *TILEDICONW* can also be used to create icons that consist solely of a title, with no special image. If the argument *ICON* is *NIL*, *TILEDICONW* creates a rectangular icon large enough to contain *TITLE*, with a border the same width as a regular window. The remaining arguments are as described above, except that a *JUST* of *TOP* or *BOTTOM* is not meaningful.

Modifying Icons

(*ICONW.TITLE* *ICON* *TITLE*) [Function]

Returns the current title of the window *ICON*, which must be a window returned by *TILEDICONW*. Additionally, if *TITLE* is non-*NIL*, makes *TITLE* be the new title of the window and repaints it accordingly. To erase the current title, make *TITLE* be a null string.

(ICONW.SHADE *WINDOW SHADE*) [Function]

Returns the current shading of the window *ICON*, which must be a window returned by *ICONW* or *TITLEDICONW*. Additionally, if *SHADE* is non-NIL, paints the texture *SHADE* on *WINDOW*. A typical use for this function is to communicate a change of state in a window that is shrunk, without reopening the window. To remove any shading, make *SHADE* be *WHITESHAE*.

Default Icons

When you shrink a window that has no *ICONFN*, the system currently creates an icon that looks like the window's title bar. You can make the system instead create titled icons by setting the global variable *DEFAULTICONFN* to the value *TEXTICON*.

(*TEXTICON WINDOW TEXT*) [Function]

Creates a titled icon window for the main window *WINDOW* containing the text *TEXT*, or the window's title if *TEXT* is NIL.

DEFAULTTEXTICON [Variable]

The value that *TEXTICON* passes to *TITLEDICONW* as its *ICON* argument. Initially NIL, which creates an unadorned rectangular window, but you can set it to a *TITLEDICON* record of your choosing if you would like default icons to have a different appearance.

Sample Icons

The file <LispUsers>StockIcons contains a collection of icons and their masks usable with *ICONW*, including:

FOLDER, *FOLDERMASK*—A file folder

PAPERICON, *PAPERICONMASK*—A sheet of paper with the top right corner turned

FILEDRAWER, *FILEDRAWERMASK*—The front of a file drawer

ENVELOPEICON, *ENVELOPEMASK*—An envelope

TITLED.FILEDRAWER—A TitledIcon of the filedrawer front (Capacity, about three lines of 10-pt. text)

TITLED.FILEFOLDER—A TitledIcon of the file folder (Capacity, about three lines of 10-pt. text)

TITLED.ENVELOPE—A TitledIcon of the envelope (Capacity, one short line of 10-pt. text)