

# PLT MrLib: Graphical Libraries Manual

---

PLT ([scheme@plt-scheme.org](mailto:scheme@plt-scheme.org))

370

Released May 2007

## Copyright notice

Copyright ©1996-2007 PLT

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Library General Public License, Version 2 published by the Free Software Foundation. A copy of the license is included in the appendix entitled “License.”

## Send us your Web links

If you use any parts or all of the PLT Scheme package (software, lecture notes) for one of your courses, for your research, or for your work, we would like to know about it. Furthermore, if you use it and publicize the fact on some Web page, we would like to link to that page. Please drop us a line at [scheme@plt-scheme.org](mailto:scheme@plt-scheme.org). Evidence of interest helps the DrScheme Project to maintain the necessary intellectual and financial support. We appreciate your help.

## Thanks

Contributors to MrLib include Mike MacHenry.

This manual was typeset using L<sup>A</sup>T<sub>E</sub>X, S<sup>I</sup>T<sub>E</sub>X, and tex2page. Some typesetting macros were originally taken from Julian Smart’s *Reference Manual for wxWindows 1.60: a portable C++ GUI toolkit*.

This manual was typeset on May 20, 2007.

# Contents

<b>1</b>	<b>MrLib</b>	<b>1</b>
<b>2</b>	<b>aligned-pasteboard.ss: Aligned Pasteboard</b>	<b>2</b>
2.1	aligned-editor-canvas%	2
2.2	aligned-editor-snip%	3
2.3	aligned-pasteboard<%>	4
2.4	aligned-pasteboard-parent<%>	4
2.5	horizontal-pasteboard%	5
2.6	stretchable-snip<%>	6
2.7	vertical-pasteboard%	7
<b>3</b>	<b>bitmap-label.ss: Bitmap Labels</b>	<b>9</b>
<b>4</b>	<b>cache-image-snip.ss: Cache-image Snip</b>	<b>10</b>
4.1	cache-image-snip%	10
4.2	Functions	12
<b>5</b>	<b>interactive-value-port.ss: Interactive Value Port</b>	<b>13</b>
<b>6</b>	<b>gif.ss: GIF and Animated GIF Writing</b>	<b>14</b>
<b>7</b>	<b>graph.ss: Graph</b>	<b>15</b>
7.1	graph-pasteboard<%>	15
7.2	graph-pasteboard-mixin	16
7.3	graph-snip<%>	16
7.4	graph-snip-mixin	17
7.5	Graph Functions	18

---

<b>8</b>	<b>include-bitmap.ss: Include Bitmap</b>	<b>20</b>
<b>9</b>	<b>name-message.ss: Name Message</b>	<b>21</b>
9.1	name-message% . . . . .	21
9.2	Name-Message Functions . . . . .	24
<b>10</b>	<b>path-dialog.ss: File/Directory Dialog</b>	<b>25</b>
10.1	path-dialog% . . . . .	25
<b>11</b>	<b>plot.ss: Plot</b>	<b>27</b>
	<b>License</b>	<b>28</b>
	<b>Index</b>	<b>32</b>

# 1. MrLib

---

The MrLib collection consists of several libraries, each of which provides a set of procedures and syntax.

To use a MrLib library, either at the top-level or within a module, import it with

```
(require (lib "libname" "mrlib"))
```

For example, to use the **graph.ss** library:

```
(require (lib "graph.ss" "mrlib"))
```

The MrLib collection provides the following libraries:

- **aligned-pasteboard.ss** – pasteboards that manage the geometry of their snips
- **bitmap-label.ss** – builds button labels from bitmaps
- **cache-image-snip.ss** – the core of the drscheme image teachpack
- **fancy-value-port.ss** – allows ports to show syntax and numbers interactively
- **graph.ss** – shows graphs in a pasteboard
- **include-bitmap.ss** – inlines bitmaps into source text for stand-alone executables
- **name-message.ss** – a control for showing a path in a pop-down menu
- **plot.ss** – library for plotting values to a dc
- **path-dialog.ss** – a platform-independent file/directory dialog
- **syntax-browser.ss** – library for browsing the structure of syntax objects
- **text-string-style-desc.ss** – normalizes snips and styles into sexps

## 2. aligned-pasteboard.ss: Aligned Pasteboard

---

To load: `(require (lib "aligned-pasteboard.ss" "mrlib"))`

The aligned-pasteboard library provides classes derived from `pasteboard%` with geometry management that mirrors that of `vertical-panel%` and `horizontal-panel%`.

- `aligned-editor-canvas%`
- `aligned-editor-snip%`
- `aligned-pasteboard<%>`
- `aligned-pasteboard-parent<%>`
- `horizontal-pasteboard%`
- `stretchable-snip<%>`
- `vertical-pasteboard%`

### 2.1 aligned-editor-canvas%

Superclass: `editor-canvas%`

Calls the `realign` method when resized.

```
- (new aligned-editor-canvas% (parent _) [(editor _)] [(style _)] [(scrolls-per-page
_) [(label _)] [(wheel-step _)] [(line-count _)] [(horizontal-inset _)] [(vertical-inset
_) [(enabled _)] [(vert-margin _)] [(horiz-margin _)] [(min-width _)] [(min-height
_) [(stretchable-width _)] [(stretchable-height _)]) => aligned-editor-canvas%
object
  parent: frame%, dialog%, panel%, or pane% object
  editor = #f: text% or pasteboard% object or #f
  style = null: list of symbols in '(no-border control-border combo no-hscroll
no-vscroll hide-hscroll hide-vscroll auto-vscroll
auto-hscroll resize-corner deleted transparent)
  scrolls-per-page = 100: exact integer in [1, 10000]
  label = #f: string (up to 200 characters) or #f
  wheel-step = 3: exact integer in [1, 10000] or #f
  line-count = #f: exact integer in [1, 1000] or #f
  horizontal-inset = 5: exact integer in [0, 1000]
  vertical-inset = 5: exact integer in [0, 1000]
  enabled = #t: boolean
  vert-margin = 0: exact integer in [0, 1000]
  horiz-margin = 0: exact integer in [0, 1000]
```

*min-width* = graphical minimum width : exact integer in [0, 10000]  
*min-height* = graphical minimum width : exact integer in [0, 10000]  
*stretchable-width* = #t : boolean  
*stretchable-height* = #t : boolean

If a canvas is initialized with #f for *editor*, install an editor later with `set-editor`.

The *style* list can contain the following flags:

- 'no-border — omits a border around the canvas
- 'control-border — gives the canvas a border that is like a `text-field%` control
- 'combo — gives the canvas a combo button that is like a `combo-field%` control; this style is intended for use with 'control-border, 'hide-hscroll, and 'hide-vscroll
- 'no-hscroll — disallows horizontal scrolling and hides the horizontal scrollbar
- 'no-vscroll — disallows vertical scrolling and hides the vertical scrollbar
- 'hide-hscroll — allows horizontal scrolling, but hides the horizontal scrollbar
- 'hide-vscroll — allows vertical scrolling, but hides the vertical scrollbar
- 'auto-hscroll — automatically hides the horizontal scrollbar when unneeded (unless 'no-hscroll or 'hide-hscroll is specified)
- 'auto-vscroll — automatically hides the vertical scrollbar when unneeded (unless 'no-vscroll or 'hide-vscroll is specified)
- 'resize-corner — leaves room for a resize control at the canvas's bottom right when only one scrollbar is visible
- 'deleted — creates the canvas as initially hidden and without affecting *parent*'s geometry; the canvas can be made active later by calling *parent*'s `add-child` method
- 'transparent — the canvas is “erased” before an update using it's parent window's background

While vertical scrolling of text editors is based on lines, horizontal scrolling and pasteboard vertical scrolling is based on a fixed number of steps per horizontal page. The *scrolls-per-page* argument sets this value.

If provided, the *wheel-step* argument is passed on to the `wheel-step` method. The default wheel step can be overridden globally though the '|MrEd:wheelStep| preference; see “Preferences” (§ in *PLT MrEd: Graphical Toolbox Manual*).

If *line-count* is not #f, it is passed on to the `set-line-count` method.

If *horizontal-inset* is not 5, it is passed on to the `horizontal-inset` method. Similarly, if *vertical-inset* is not 5, it is passed on to the `vertical-inset` method.

For information about the *enabled* argument, see `window<%>`. For information about the *horiz-margin* and *vert-margin* arguments, see `subarea<%>`. For information about the *min-width*, *min-height*, *stretchable-width*, and *stretchable-height* arguments, see `area<%>`.

## 2.2 aligned-editor-snip%

Superclass: `editor-snip%`

Calls the `realign` method when resized.

```
- (new aligned-editor-snip% [(editor -)] [(with-border? -)] [(left-margin -)]
  [(top-margin -)] [(right-margin -)] [(bottom-margin -)] [(left-inset -)] [(top-inset
-)] [(right-inset -)] [(bottom-inset -)] [(min-width -)] [(max-width -)] [(min-height
-)] [(max-height -)]) => aligned-editor-snip% object
  editor = #f : text% object or #f
  with-border? = #t : boolean
  left-margin = 5 : exact non-negative integer
  top-margin = 5 : exact non-negative integer
  right-margin = 5 : exact non-negative integer
```

```

bottom-margin = 5 : exact non-negative integer
left-inset = 1 : exact non-negative integer
top-inset = 1 : exact non-negative integer
right-inset = 1 : exact non-negative integer
bottom-inset = 1 : exact non-negative integer
min-width = 'none : non-negative real number or 'none
max-width = 'none : non-negative real number or 'none
min-height = 'none : non-negative real number or 'none
max-height = 'none : non-negative real number or 'none

```

If *editor* is non-#f, then it will be used as the editor contained by the snip. See also [set-editor](#).

If *with-border?* is not #f, then a border will be drawn around the snip. The editor display will be inset in the snip area by the amounts specified in the *-margin* arguments. The border will be drawn with an inset specified by the *-inset* arguments.

See [get-inset](#) and [get-margin](#) for information about the inset and margin arguments.

## 2.3 aligned-pasteboard<%>

get-aligned-min-height

The minimum height an aligned-pasteboard can be and still fit the heights of all of its children.

```
- (send an-aligned-pasteboard get-aligned-min-height) => number
```

get-aligned-min-width

The minimum width an aligned-pasteboard can be and still fit the widths of all of its children.

```
- (send an-aligned-pasteboard get-aligned-min-width) => number
```

realign

```
- (send an-aligned-pasteboard realign width height) => void
  width: nonnegative
  height: nonnegative
```

Realigns the children inside the [aligned-pasteboard<%>](#) to the given width and height.

```
- (send an-aligned-pasteboard realign) => void
```

Realigns the children inside the aligned-pasteboard; the previously allotted width and height.

set-aligned-min-sizes

Calculates the minimum width and height of the of the pasteboard based on children's min-sizes and stores it for later retrieval via the getters.

```
- (send an-aligned-pasteboard set-aligned-min-sizes) => number
```

## 2.4 aligned-pasteboard-parent<%>

This interface must be implemented by any class who's editor is an [aligned-pasteboard<%>](#).



set-aligned-min-sizes

- (send *an-aligned-pasteboard-parent* set-aligned-min-sizes) ⇒ void

## 2.5 horizontal-pasteboard%

Superclass: `pasteboard%`

Implements: `aligned-pasteboard<%>`

- (new horizontal-pasteboard% ) ⇒ `horizontal-pasteboard%` object

The editor will not be displayed until it is attached to an `editor-canvas%` object or some other `display`.

A new `keymap%` object is created for the new editor. See also `get-keymap` and `set-keymap`.

A new `style-list%` object is created for the new editor. See also `get-style-list` and `set-style-list`.

after-delete

Called after a snip is deleted from the editor (and after the `display` is refreshed; use `on-delete` and `begin-edit-sequence` to avoid extra refreshes when `after-delete` modifies the editor).

See also `can-delete?` and `on-edit-sequence`.

No internals locks are set when this method is called.

- (send *an-horizontal-pasteboard* after-delete *snip*) ⇒ void  
*snip*: `snip%` object

after-insert

Called after a snip is inserted into the editor (and after the `display` is refreshed; use `on-insert` and `begin-edit-sequence` to avoid extra refreshes when `after-insert` modifies the editor).

See also `can-insert?` and `on-edit-sequence`.

No internals locks are set when this method is called.

- (send *an-horizontal-pasteboard* after-insert *snip* *before* *x* *y*) ⇒ void  
*snip*: `snip%` object  
*before*: `snip%` object or #f  
*x*: real number  
*y*: real number

after-reorder

Called before a snip is moved in the pasteboard's front-to-back snip order (and after the `display` is refreshed; use `on-reorder` and `begin-edit-sequence` to avoid extra refreshes when `after-reorder` modifies the editor).

See also [can-reorder?](#) and [on-edit-sequence](#).

No internals locks are set when this method is called.

```
- (send an-horizontal-pasteboard after-reorder snip to-snip before?) ⇒ void
  snip: snip% object
  to-snip: snip% object
  before?: boolean
```

resized

Called (indirectly) by snips within the editor: it forces a recalculation of the display information in which the specified snip has changed its size.

```
- (send an-horizontal-pasteboard resized snip redraw-now?) ⇒ void
  snip: snip% object
  redraw-now?: boolean
```

## 2.6 stretchable-snip<%>

This interface must be implemented by any snip class whose objects will be stretchable when inserted into an [aligned-pasteboard<%>](#).

get-aligned-min-height

The minimum height that the snip can be resized to

```
- (send a-stretchable-snip get-aligned-min-height) ⇒ number
```

get-aligned-min-width

The minimum width that the snip can be resized to.

```
- (send a-stretchable-snip get-aligned-min-width) ⇒ number
```

stretchable-height

Whether or not the snip can be stretched in the Y dimension

```
- (send a-stretchable-snip stretchable-height) ⇒ boolean
```

stretchable-width

Whether or not the snip can be stretched in the X dimension

```
- (send a-stretchable-snip stretchable-width) ⇒ boolean
```

## 2.7 vertical-pasteboard%

Superclass: `pasteboard%`

Implements: `aligned-pasteboard<%>`

- (new vertical-pasteboard% ) ⇒ `vertical-pasteboard%` object

The editor will not be displayed until it is attached to an `editor-canvas%` object or some other `display`.

A new `keymap%` object is created for the new editor. See also `get-keymap` and `set-keymap`.

A new `style-list%` object is created for the new editor. See also `get-style-list` and `set-style-list`.

`after-delete`

Called after a snip is deleted from the editor (and after the `display` is refreshed; use `on-delete` and `begin-edit-sequence` to avoid extra refreshes when `after-delete` modifies the editor).

See also `can-delete?` and `on-edit-sequence`.

No internals locks are set when this method is called.

- (send *a-vertical-pasteboard* after-delete *snip*) ⇒ void  
  *snip*: `snip%` object

`after-insert`

Called after a snip is inserted into the editor (and after the `display` is refreshed; use `on-insert` and `begin-edit-sequence` to avoid extra refreshes when `after-insert` modifies the editor).

See also `can-insert?` and `on-edit-sequence`.

No internals locks are set when this method is called.

- (send *a-vertical-pasteboard* after-insert *snip* before *x* *y*) ⇒ void  
  *snip*: `snip%` object  
  *before*: `snip%` object or #f  
  *x*: real number  
  *y*: real number

`after-reorder`

Called before a snip is moved in the pasteboard's front-to-back snip order (and after the `display` is refreshed; use `on-reorder` and `begin-edit-sequence` to avoid extra refreshes when `after-reorder` modifies the editor).

See also `can-reorder?` and `on-edit-sequence`.

No internals locks are set when this method is called.

- (send *a-vertical-pasteboard* after-reorder *snip* to-*snip* before?) ⇒ void  
  *snip*: `snip%` object

```
to-snip: snip% object  
before?: boolean
```

resized

Called (indirectly) by snips within the editor: it forces a recalculation of the display information in which the specified snip has changed its size.

```
- (send a-vertical-pasteboard resized snip redraw-now?) ⇒ void  
  snip: snip% object  
  redraw-now?: boolean
```

### 3. `bitmap-label.ss`: Bitmap Labels

---

To load: `(require (lib "bitmap-label.ss" "mrlib"))`

`(make-bitmap-label string bitmap-or-path [font])` PROCEDURE

Constructs a bitmap label suitable for use a button that contains the image named by *bitmap-or-path* followed by the text in *string*.

`(bitmap-label-maker string bitmap-or-path)` PROCEDURE

OBSOLETE. Use `make-bitmap-label` instead.

This function takes a string and `bitmap%` (or filename) and returns a function that takes a container and generates a bitmap. The container determines the font that is used for the bitmap label. The result is a bitmap (with a mask) that is suitable for use as a `button%` label.

## 4. cache-image-snip.ss: Cache-image Snip

---

To load: `(require (lib "cache-image-snip.ss" "mrlib"))`

This library is the core data structure for the image.ss teachpack. Images in the image.ss library are instances of the `cache-image-snip%` class.

The library also defines a new type, `argb`, that represents a bitmap, but with alpha values. It has a maker, two selectors, and a predicate.

- `cache-image-snip%`
- **Functions**

### 4.1 cache-image-snip%

Superclass: `snip%`

- `(make-object cache-image-snip%)` ⇒ `cache-image-snip%` object  
Creates a plain snip of length 1 with the "Basic" style of the-style-list.

`get-argb`

- `(send a-cache-image-snip get-argb)` ⇒ `argb`  
Returns a pixel array for this image, forcing it to be computed.

`get-argb-proc`

- `(send a-cache-image-snip get-argb-proc)` ⇒ `(argb number number .-i . void)`  
Returns a procedure that fills in an `argb` with the contents of this image at the given offset

`get-argb/no-compute`

- `(send a-cache-image-snip get-argb/no-compute)` ⇒ `(union #f argb)`  
Returns a pixel array for this image or `#f` if it has not been computed yet.

`get-bitmap`

- `(send a-cache-image-snip get-bitmap)` ⇒ `bitmap`  
Builds (if not yet built) a bitmap corresponding to this snip and returns it.

get-dc-proc

- (send *a-cache-image-snip* get-dc-proc) ⇒ (union #f ((is-a?/c dc<\*>) int[dx] int[dy] -i void))

Either returns false, or a procedure that draws the contents of this snip into a dc.

get-pinhole

- (send *a-cache-image-snip* get-pinhole) ⇒ (values number number)

returns the pinhole coordinates for this image, counting from the top-left of the image.

get-size

- (send *a-cache-image-snip* get-size) ⇒ (values number number)

returns the width and height for the image.

## 4.2 Functions

snip-class SNIPCLASS

This snipclass is used for saved cache image snips.

(make-argb (*vector of rational [between 0 & 255]*) *int*) CONSTRUCTOR

Constructs a new argb value. The vector has four entries for each pixel, an alpha, red, green, and blue value. The int specifies the width of the image; the height is the size of the vector, divided by 4, divided by the width.

(argb-vector *argb*) SELECTOR

Extracts the vector from an argb

(argb-width *argb*) SELECTOR

Extracts the width from an argb.

(argb? *any*) PREDICATE

Tests if its argument is an argb.

(overlay-bitmap *dx dy color-bitmap mask-bitmap*) PROCEDURE

Builds a new argb after overlaying color-bitmap (with masking based on mask-bitmap) onto argb at (dx, dy) from the top-left.

(build-bitmap *draw w h*) PROCEDURE

Builds a bitmap with width *w* and height *h*, using the procedure *draw*. The procedure should accept one argument, `dc<%>`, and return void.

(flatten-bitmap *bitmap*) PROCEDURE

Builds a new bitmap that flattens the original bitmap with its mask, producing a bitmap that has no mask, and looks the way that bitmap would draw (when drawn with the mask).

(argb->cache-image-snip *argb number number*) PROCEDURE

Builds a new `cache-image-snip%` based on the contents of argb, using the two numbers as the pinhole.

(argb->bitmap *argb*) PROCEDURE

Builds a bitmap that draws the same way as argb; the alpha pixels are put into the bitmap's get-loaded-mask bitmap.



## 5. interactive-value-port.ss: Interactive Value Port

---

To load: `(require (lib "interactive-value-port.ss" "mrlib"))`

`(set-interactive-display-handler port)`

PROCEDURE

Sets the port-display-handler for the given port so that when it encounters these values:

- exact, real, non-integral numbers
- syntax objects

it uses `write-special` to send snips to the port, instead of those values. Otherwise, it behaves like the default port-display-handler.

In order to show values embedded in lists and other compound object, it uses `pretty-print`.

`(set-interactive-write-handler port)`

PROCEDURE

Same as `set-interactive-display-handler`, except that it sets the port-print-handler.

`(set-interactive-print-handler port)`

PROCEDURE

Same as `set-interactive-display-handler`, except that it sets the port-display-handler.

## 6. gif.ss: GIF and Animated GIF Writing

---

To load: `(require (lib "gif.ss" "mrlib"))`

The **gif.ss** library provides functions for writing bitmap objects to GIF files.

`(write-gif bitmap filename)`

PROCEDURE

Writes the given *bitmap* to *filename* as a GIF image, where *bitmap* is either an instance of `bitmap%` or a thunk (to be called just once) that generates such an object. If the bitmap uses more than 256 colors, it is automatically quantized using a simple algorithm. If the bitmap has a mask bitmap, it is used to determine transparent pixels in the generated GIF image.

`(write-animated-gif bitmap-list delay filename [#:one-at-a-time? one-at-a-time?])`  
PROCEDURE

Writes the bitmaps in *bitmap-list* to *filename* as an animated GIF. The *bitmap-list* list can contain a mixture of `bitmap%` objects and thunks (each called just once) that produce `bitmap%` objects. The *delay* argument is the amount of time in 1/100s of a second to wait between transitions.

If *one-at-a-time?* is `#f` (the default), then the content of all images is collected and quantized at once, to produce a single colortable; a drawback to this approach is that it uses more memory, and it allows less color variation among animation frames. Even when *one-at-a-time?* is false, the result of each thunk in *bitmap-list* is converted to a byte-string one at a time (which helps avoid bitmap-count limits under Windows).

If *one-at-a-time?* is true, then the bitmaps are quantized and written to the file one at a time; that is, for each thunk in *bitmap-list*, its result is written and discarded before another thunk is called. A drawback to this approach is that a separate colortable is written for each frame in the animation, which can make the resulting file large.

## 7. graph.ss: Graph

---

To load: `(require (lib "graph.ss" "mrlib"))`

The **graph.ss** library provides a graph drawing toolkit, built out of MrEd's `pasteboard%`s.

- `graph-pasteboard<%>`
- `graph-pasteboard-mixin%`
- `graph-snip<%>`
- `graph-snip-mixin%`
- **Graph Functions**

### 7.1 graph-pasteboard<%>

`get-arrowhead-params`

- `(send a-graph-pasteboard get-arrowhead-params) ⇒ (values number number number)`

Returns the current settings for the arrowhead's drawing.

`on-mouse-over-snips`

- `(send a-graph-pasteboard on-mouse-over-snips a-los) ⇒ void`  
`a-los: (listof (is-a?/c snip%))`

This method is called when the mouse passes over any snips in the editor. It is only called when the list of snips under the editor changes (ie, if the mouse moves, but remains over the same list of snips, the method is not called). Also, this method is called with the empty list if the mouse leaves the pasteboard.

`set-arrowhead-params`

- `(send a-graph-pasteboard set-arrowhead-params angle-width short-side long-size)`  
`⇒ void`  
`angle-width: number`  
`short-side: number`  
`long-size: number`

Sets drawing parameters for the arrowhead. The first is the angle of the arrowhead's point, in radians. The second is the length of the outside line of the arrowhead and the last is the distance from the arrowhead's point to the place where the arrowhead comes together.

set-draw-arrow-heads?

```
- (send a-graph-pasteboard set-draw-arrow-heads? draw-arrow-heads?) => void
  draw-arrow-heads?: boolean
```

Sets a boolean controlling whether or not arrow heads are drawn on the edges between nodes.

(Does not affect self-links, only links between two different nodes.)

## 7.2 graph-pasteboard-mixin

Domain: (class->interface `pasteboard%`)

Implements: `graph-pasteboard<%>`

This mixin overrides many methods to draw lines between `graph-snip<%>` that it contains.

```
- init args:
```

The editor will not be displayed until it is attached to an `editor-canvas%` object or some other `display`.

A new `keymap%` object is created for the new editor. See also `get-keymap` and `set-keymap`.

A new `style-list%` object is created for the new editor. See also `get-style-list` and `set-style-list`.

## 7.3 graph-snip<%>

add-child

```
- (send a-graph-snip add-child child) => void
  child: (is-a?/c graph-snip<%>)
```

Adds a child of this snip. Instead of calling this method, consider using the `add-links` function provided from this library.

add-parent

Adds a parent of this snip. Instead of calling this method, consider using `add-links` function provided from this library.

```
- (send a-graph-snip add-parent parent) => void
  parent: (is-a?/c graph-snip<%>)
```

```
- (send a-graph-snip add-parent parent mouse-over-pen mouse-off-pen mouse-over-brush
  mouse-off-brush) => void
  parent: (is-a?/c graph-snip<%>)
  mouse-over-pen: (union false? (is-a?/c pen%))
  mouse-off-pen: (union false? (is-a?/c pen%))
  mouse-over-brush: (union false? (is-a?/c brush%))
  mouse-off-brush: (union false? (is-a?/c brush%))
```

get-children

```
- (send a-graph-snip get-children) ⇒ (listof snip%)
```

returns a list of snips that implement *graph-snip<%>*. Each of these snips will have a line drawn from it, pointing at this snip.

get-parents

```
- (send a-graph-snip get-parents) ⇒ (listof graph-snip<%>)
```

Returns a list of snips that implement *graph-snip<%>*. Each of these snips will have a line drawn to it, starting from this snip.

remove-child

```
- (send a-graph-snip remove-child child) ⇒ void
  child: (is-a?/c graph-snip<%>)
```

Removes a child snip from this snip. Be sure to remove this snip as a parent from the argument, too.

remove-parent

```
- (send a-graph-snip remove-parent parent) ⇒ void
  parent: (is-a?/c graph-snip<%>)
```

removes a parent snip from this snip. Be sure to remove this snip as a child from the argument, too.

## 7.4 graph-snip-mixin

Domain: (class->interface *snip%*)

Implements: *graph-snip<%>*

```
- init args:
```

Creates a plain snip of length 1 with the "Basic" style of *the-style-list*.

## 7.5 Graph Functions

```
(add-links graph-snip<%> graph-snip<%>) PROCEDURE
```

```
(add-links
graph-snip<%>
graph-snip<%>
(union false? (is-a?/c pen%))
(union false? (is-a?/c pen%))
(union false? (is-a?/c brush%))
(union false? (is-a?/c brush%))) PROCEDURE
```

```
(add-links
graph-snip<%>
graph-snip<%>
(union false? (is-a?/c pen%))
(union false? (is-a?/c pen%))
(union false? (is-a?/c brush%))
(union false? (is-a?/c brush%))
(union false? string)) PROCEDURE
```

```
(add-links
graph-snip<%>
graph-snip<%>
(union false? (is-a?/c pen%))
(union false? (is-a?/c pen%))
(union false? (is-a?/c brush%))
(union false? (is-a?/c brush%))
number
number) PROCEDURE
```

```
(add-links
graph-snip<%>
graph-snip<%>
(union false? (is-a?/c pen%))
(union false? (is-a?/c pen%))
(union false? (is-a?/c brush%))
(union false? (is-a?/c brush%))
number
number
(union false? string)) PROCEDURE
```

The `add-links` function connects a parent snip to a child snip in the same pasteboard.

When called with two arguments, connects the snips using a blue/purple color scheme for the links. The first snip is the parent and the second snip is the child.

When called with six arguments, uses the two pens and brushes for the color scheme. The first pen and the first brush are used when the mouse cursor is over the snip (or a child or parent) and the second pen and brush are used when the mouse cursor is not over the snip. The brush is used to draw inside the arrow head and the pen is used to draw the border of the arrowhead and the line connecting the two snips.

When called with a final string argument, the string is used as a label on the edge.

When called with two final number arguments, they are dx and dy offsets for the head and the tail of the arrow, first dx and then dy.

When called with two numbers and a string as the final three arguments, the numbers and strings behave as they do separately.

```
(add-links/text-colors
graph-snip<%>
graph-snip<%>
(union false? (is-a?/c pen%))
(union false? (is-a?/c pen%))
(union false? (is-a?/c brush%))
(union false? (is-a?/c brush%))
(union false? (is-a?/c color%))
(union false? (is-a?/c color%))
number
number
```

```
(union false? string))
```

PROCEDURE

Behaves like *add-links*, except that seventh and eighth arguments specify colors for the text on the label (the arguments preceding and following are treated like the last case of *add-links*).

## 8. include-bitmap.ss: Include Bitmap

---

To load: `(require (lib "include-bitmap.ss" "mrlib"))`

The **include-bitmap.ss** library provides a `include-bitmap` form that takes a filename containing a bitmap and “inlines” the bitmap into the program. The advantage of inlining the bitmap is that a stand-alone executable can be created that contains the bitmap and does not refer to the original image file.

`(include-bitmap file-spec [type-expr])` SYNTAX

The *file-spec* is the same as for MzLib’s `include` form: a path string, a `build-path` form, or a `lib` form. The *type-expr* should produce `'unknown`, `'unknown/mask`, etc., and the default is `'unknown/mask`.

`(include-bitmap/relative-to source file-spec [type-expr])` SYNTAX

Analogous to `include-at/relative-to`, though only a source is needed (no context).



## 9. name-message.ss: Name Message

---

To load: `(require (lib "name-message.ss" "mrlib"))`

- `name-message%`
- `Name-Message Functions`

### 9.1 name-message%

Superclass: `canvas%`

A `name-message%` control displays a filename that the user can click to show the filename's path and select one of the enclosing directories. Override the `on-choose-directory` method to handle the user's selection.

```
- (new name-message% (parent _) [(style _)] [(paint-callback _)] [(label _)] [(gl-config
_)]) [(enabled _)] [(vert-margin _)] [(horiz-margin _)] [(min-width _)] [(min-height
_)]) [(stretchable-width _)] [(stretchable-height _)]) => name-message% object
parent: frame%, dialog%, panel%, or pane% object
style=null: list of symbols in '(border control-border combo vscroll hscroll
resize-corner gl deleted no-autoclear transparent)
paint-callback=void: procedure of two arguments: a canvas% object and a dc<%> object
label=#f: string (up to 200 characters) or #f
gl-config=#f: gl-config% object or #f
enabled=#t: boolean
vert-margin=0: exact integer in [0, 1000]
horiz-margin=0: exact integer in [0, 1000]
min-width=graphical minimum width: exact integer in [0, 10000]
min-height=graphical minimum width: exact integer in [0, 10000]
stretchable-width=#t: boolean
stretchable-height=#t: boolean
```

The `style` argument indicates one or more of the following styles:

- 'border — gives the canvas a thin border
- 'control-border — gives the canvas a border that is like a `text-field%` control
- 'combo — gives the canvas a combo button that is like a `combo-field%` control; this style is intended for use with 'control-border and not with 'hscroll or 'vscroll
- 'hscroll — enables horizontal scrolling (initially visible but inactive)
- 'vscroll — enables vertical scrolling (initially visible but inactive)
- 'resize-corner — leaves room for a resize control at the canvas's bottom right when only one scrollbar is visible
- 'gl — *obsolete* (every canvas is an OpenGL context where supported)
- 'deleted — creates the canvas as initially hidden and without affecting `parent`'s geometry; the canvas can be made active later by calling `parent`'s `add-child` method
- 'no-autoclear — prevents automatic erasing of the canvas before calls to `on-paint`

- 'transparent — the canvas is automatically “erased” before an update using its parent window’s background; the result is undefined if this flag is combined with 'no-autoclear

The 'hscroll and 'vscroll styles create a canvas with an initially inactive scrollbar. The scrollbars are activated with either `init-manual-scrollbars` or `init-auto-scrollbars`, and they can be hidden and re-shown with `show-scrollbars`.

The `paint-callback` argument is called by the default `on-paint` method, using the canvas and the DC returned by `get-dc` as the argument.

The `label` argument names the canvas for `get-label`, but it is not displayed with the canvas.

The `gl-config` argument determines properties of an OpenGL context for this canvas, as obtained through the canvas’s drawing context. See also `get-dc` and `get-gl-context in dc<%>`.

For information about the `enabled` argument, see `window<%>`. For information about the `horiz-margin` and `vert-margin` arguments, see `subarea<%>`. For information about the `min-width`, `min-height`, `stretchable-width`, and `stretchable-height` arguments, see `area<%>`.

on-choose-directory

- (send *a-name-message* on-choose-directory *dir*) ⇒ void  
*dir*: path

Called when one of the popup menu items is chosen. The argument is a represents the selected directory.

on-event

Called when the canvas receives a mouse event. See also “Mouse and Keyboard Events” (§ in *PLT MrEd: Graphical Toolbox Manual*), noting in particular that certain mouse events can get dropped.

- (send *a-name-message* on-event *event*) ⇒ void  
*event*: `mouse-event%` object

Handles the click by popping up a menu or message.

on-paint

Called when the canvas is exposed or resized so that the image in the canvas can be repainted.

When `on-paint` is called in response to a system expose event and only a portion of the canvas is newly exposed, any drawing operations performed by `on-paint` are clipped to the newly-exposed region; however, the clipping region as reported by `get-clipping-region` does not change.

- (send *a-name-message* on-paint) ⇒ void  
 Draws the control’s current message.

set-hidden?

Calling this method with #f causes the name message to become invisible and to stop responding to mouse movements.

Calling it with #t restores its visibility and makes it respond to mouse movements again.

- (send *a-name-message* set-hidden?) ⇒ void

set-message

Sets the label for the control.

```
- (send a-name-message set-message file-name? msg) ⇒ void
  file-name? : boolean
  msg : path string
```

If *file-name?* is #t, *msg* is treated like a pathname, and a click on the name-message control creates a popup menu to open a get-file dialog.

If *file-name?* is #f, *msg* is treated as a label string. Clicking on the name-message control pops up a dialog saying that there is no file name until the file is saved.

## 9.2 Name-Message Functions

(*calc-button-min-sizes dc string*) PROCEDURE

Calculates the minimum width and height of a button label (when drawn with *draw-frame-button-label*). Returns two values: the width and height.

(*draw-frame-button-label dc string width height inverted*) PROCEDURE

Draws a button label like the one for the (*define ...*) and filename buttons in the top-left corner of the DrScheme frame. Use this function to draw similar buttons.

The basic idea is to create a canvas object whose *on-paint* method is overridden to call this function. The *dc* argument should be canvas's drawing context, and *string* should be the string to display on the button. The *width* and *height* arguments should be the width and height of the button, and *inverted?* should be *#t* when the button is pressed.

See *calc-button-min-sizes* for help calculating the min sizes of the button.

## 10. path-dialog.ss: File/Directory Dialog

---

To load: `(require (lib "path-dialog.ss" "mrlib"))`

- `path-dialog%`

### 10.1 path-dialog%

Superclass: `dialog%`

The `path-dialog%` class implements a platform-independent file/directory dialog. The dialog is the same as the dialog under X for MrEd's `get-file`, `put-file`, `get-directory`, and `get-file-list` procedures, but considerable extra functionality is available through the class.

```
- (new path-dialog% [(label _)] [(message _)] [(parent _)] [(directory _)] [(filename
-) ] [(put? _)] [(dir? _)] [(existing? _)] [(new? _)] [(multi? _)] [(can-mkdir?
-) ] [(filters _)] [(show-file? _)] [(show-dir? _)] [(ok? _)] [(guard _)]) =>
path-dialog% object
  label = #f : string (up to 200 characters) or #f
  message = #f : string (up to 200 characters) or #f
  parent = #f : frame% or dialog% object or #f
  directory = #f : path or #f
  filename = #f : path or #f
  put? = #f : boolean
  dir? = #f : boolean
  existing? = (not put?) : boolean
  new? = #f : boolean
  multi? = #f : boolean
  can-mkdir? = put? : boolean
  filters = #t : list of (list filter-name filter-glob), #f, or #t
  show-file? = #f : path-to-boolean procedure or #f
  show-dir? = #f : path-to-boolean procedure or #f
  ok? = #f : path-to-boolean procedure or #f
  guard = #f : procedure or #f
```

The `label` argument is the dialog's title string. If `label` is #f, the default is based on other field values.

The `message` argument is a prompt message to show at the top of the dialog. If it is #f, no prompt line.

The `parent` argument is the parent frame or dialog, if any, for this dialog.

The `directory` argument specifies the dialog's initial directory. If it is #f, the initial directory is the last directory that was used by the user (or the current directory on first use).

The `filename` argument provides an initial filename text, if any.

If `put?` is true, the dialog operates in choose-file-to-write mode (and warn the user if choosing an existing name).

If *dir?* is true, the dialog operates in directory-choice mode.

If *existing?* is true, the user must choose an existing file.

If *new?* is true, the user must choose a non-existent path. Providing both *new?* and *existing?* as true triggers an exception.

If *multi?* is true, the dialog allows selection of multiple paths.

If *can-mkdir?* is true, the dialog includes a button for the user to create a new directory.

The *filters* argument is one of:

- (*list-of* (*list filter-name filter-glob*)) — a list of pattern names (e.g., “Scheme Files”) and glob patterns (e.g., “\*.scm;\*.ss”). Any list (including an empty list) enables a filter box for the user to enter glob patterns, and the given list of choices is available in a combo-box drop-down menu. Glob patterns are the usual Unix ones (see *glob->regex*), and a semicolon can be used to allow multiple patterns.
- #f — no patterns and no filter input box.
- #t — use a generic “All” filter, which is “\*. \*” on Windows and “\*” on other platforms.

The *show-file?* predicate is used to filter file paths that are shown in the dialog. The predicate is applied to the file name as a string while the *current-directory* parameter is set. This predicate is intended to be a lightweight filter for choosing which names to display.

The *show-dir?* predicate is similar, but for directories instead of files.

The *ok?* predicate is used in a similar fashion to the *show-file?* and *show-dir?* predicate, but it is used to determine whether the OK button should be enabled when a file or directory is selected (so it need not be as lightweight as the other predicates).

The *guard* procedure is a generic verifier for the dialog’s final result, as produced by the *run* method. It receives the result that is about to be returned (which can be a list in a multi-selection dialog), and can return a different value (any value) instead. If it throws an exception, an error dialog is shown, and the dialog interaction continues (so it can be used to verify results without dismissing the dialog). This procedure can also raise *void*, in which case the dialog remains without an error message.

*run*

This method shows the dialog and returns the selected result. If a *guard* procedure is not supplied when the dialog is created, then the result is either a path or a list of paths (and the latter only when *schememulti?* is true when the dialog is created). If a *guard* procedure is supplied, its result determines the result of this method.

```
- (send a-path-dialog run) ⇒ value
```

## 11. plot.ss: Plot

---

To load: `(require (lib "plot.ss" "mrlib"))`

The **plot.ss** library provides a simple tool for plotting data values to a device context.

Two structures are provided: `data-set` and `plot-setup`.

A `data-set` value includes the following fields (which should be supplied on order to make-`data-set`):

- `points`: `(listof (is-a?/c point%))` — the data values to plot.
- `connected?`: `any?` — indicates whether the points are connected by a line.
- `pen`: `(is-a?/c pen%)` — the drawing pen for plotting points/lines.
- `min-x`: `number?`, `max-x`: `number?`, `min-y`: `number?`, and `max-y`: `number?` — the to plot the points, in drawing-context coordinates.

A `plot-setup` value includes the following fields (which should be supplied on order to plot-`setup`):

- `axis-label-font`: `(is-a?/c font%)` — the font for drawing axis labels.
- `axis-number-font`: `(is-a?/c font%)` — the font for drawing axis numbers.
- `axis-pen`: `(is-a?/c pen%)` — the pen for drawing the axes.
- `grid?`: `any?` — whether to draw a grid at axis markings.
- `grid-pen`: `(is-a?/c pen%)` the pen for drawing the grid (if any).
- `x-axis-marking`: `(listof number?)` — locations for marks on the x-axis.
- `y-axis-marking`: `(listof number?)` — locations for marks on the y-axis.
- `x-axis-label`: `string?` — the x-axis label.
- `y-axis-label`: `string?` — the y-axis label.

`(plot dc data-set-list plot-setup)`

PROCEDURE

Draws the data-sets in `data-set-list` into the given `dc`. Uses drawing-context coordinates in data-sets that will accommodate all of the data sets.

# License

## GNU Library General Public License

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.

675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries



themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a “work based on the library” and a “work that uses the library”. The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called “this License”). Each licensee is addressed as “you”.

A “library” means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The “Library”, below, refers to any such software library or work which has been distributed under these terms. A “work based on the Library” means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term “modification”).

“Source code” for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library’s complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the “work that uses the Library” must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.  
Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.
14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

# Index

add-child, 16  
add-links, 18  
add-links/text-colors, 19  
add-parent, 16  
after-delete, 5, 7  
after-insert, 5, 7  
after-reorder, 5, 7  
aligned-editor-canvas%, 2  
aligned-editor-snip%, 3  
aligned-pasteboard-parent<%>, 4  
**aligned-pasteboard.ss**, 2  
aligned-pasteboard<%>, 4  
argb->bitmap, 12  
argb->cache-image-snip, 12  
argb-vector, 12  
argb-width, 12  
argb?, 12  
'auto-hscroll, 2  
'auto-vscroll, 2

bitmap-label-maker, 9  
**bitmap-label.ss**, 9  
'border, 21  
bottom-inset, 3  
bottom-margin, 3  
build-bitmap, 12

**cache-image-snip.ss**, 10  
cache-image-snip%, 10  
calc-button-min-sizes, 24  
can-mkdir?, 25  
canvas  
    scroll bars, 3  
canvas%, 21  
'combo, 2, 21  
'control-border, 2, 21

data-set, 27  
data-set-connected?, 27  
data-set-max-x, 27  
data-set-max-y, 27  
data-set-min-x, 27  
data-set-min-y, 27  
data-set-pen, 27  
data-set-points, 27  
data-set?, 27  
'deleted, 2, 21  
dialog%, 25  
dir?, 25

directory, 25  
draw-frame-button-label, 24

editor, 2, 3  
editor-canvas%, 2  
editor-snip%, 3  
editors  
    hooks, 5, 7  
enabled, 2, 21  
existing?, 25

filename, 25  
filters, 25  
flatten-bitmap, 12

get-aligned-min-height, 4, 6  
get-aligned-min-width, 4, 6  
get-argb, 10  
get-argb-proc, 10  
get-argb/no-compute, 10  
get-arrowhead-params, 15  
get-bitmap, 10  
get-children, 17  
get-dc-proc, 11  
get-parents, 17  
get-pinhole, 11  
get-size, 11  
**gif.ss**, 14  
'gl, 21  
gl-config, 21  
graph-pasteboard-mixin, 16  
graph-pasteboard<%>, 15  
graph-snip-mixin, 17  
graph-snip<%>, 16  
**graph.ss**, 15  
guard, 25

'hide-hscroll, 2  
'hide-vscroll, 2  
horiz-margin, 2, 21  
horizontal-inset, 2  
horizontal-pasteboard%, 5  
'hscroll, 21

include-bitmap, 20  
**include-bitmap.ss**, 20  
include-bitmap/relative-to, 20  
**interactive-value-port.ss**, 13

keymaps

- in an editor, 5, 7, 16
- label, 2, 21, 25
- left-inset, 3
- left-margin, 3
- line-count, 2
- make-argb, 12
- make-bitmap-label, 9
- make-data-set, 27
- make-plot-setup, 27
- max-height, 3
- max-width, 3
- message, 25
- min-height, 2, 3, 21
- min-width, 2, 3, 21
- '|MrEd:wheelStep|, 3
- multi?, 25
- name-message.ss**, 21
- name-message%, 21
- new?, 25
- 'no-autoclear, 21
- 'no-border, 2
- 'no-hscroll, 2
- 'no-vscroll, 2
- ok?, 25
- on-choose-directory, 22
- on-event, 22
- on-mouse-over-snips, 15
- on-paint, 22
- overlay-bitmap, 12
- paint-callback, 21
- parent, 2, 21, 25
- pasteboard%, 5, 7
- path-dialog.ss**, 25
- path-dialog%, 25
- plot, 27
- plot-setup, 27
- plot-setup-axis-label-font, 27
- plot-setup-axis-number-font, 27
- plot-setup-axis-pen, 27
- plot-setup-grid-pen, 27
- plot-setup-grid?, 27
- plot-setup-x-axis-label, 27
- plot-setup-x-axis-marking, 27
- plot-setup-y-axis-label, 27
- plot-setup-y-axis-marking, 27
- plot-setup?, 27
- plot.ss**, 27
- put?, 25
- realign, 4
- remove-child, 17
- remove-parent, 17
- 'resize-corner, 2, 21
- resized, 6, 8
- right-inset, 3
- right-margin, 3
- run, 26
- scrolls-per-page, 2
- set-aligned-min-sizes, 4, 5
- set-arrowhead-params, 15
- set-data-set-connected?!, 27
- set-data-set-max-x!, 27
- set-data-set-max-y!, 27
- set-data-set-min-x!, 27
- set-data-set-min-y!, 27
- set-data-set-pen!, 27
- set-data-set-points!, 27
- set-draw-arrow-heads?, 16
- set-hidden?, 22
- set-interactive-display-handler, 13
- set-interactive-print-handler, 13
- set-interactive-write-handler, 13
- set-message, 23
- set-plot-setup-axis-label-font!, 27
- set-plot-setup-axis-number-font!, 27
- set-plot-setup-axis-pen!, 27
- set-plot-setup-grid-pen!, 27
- set-plot-setup-grid?!, 27
- set-plot-setup-x-axis-label!, 27
- set-plot-setup-x-axis-marking!, 27
- set-plot-setup-y-axis-label!, 27
- set-plot-setup-y-axis-marking!, 27
- show-dir?, 25
- show-file?, 25
- snip%, 10
- stretchable-height, 2, 21
- stretchable-height, 6
- stretchable-snip<%>, 6
- stretchable-width, 2, 21
- stretchable-width, 6
- struct:data-set, 27
- struct:plot-setup, 27
- style, 2, 21
- style lists
  - in an editor, 5, 7, 16
- the-style-list, 10, 17
- top-inset, 3
- top-margin, 3
- 'transparent, 2, 21
- vert-margin, 2, 21
- vertical-inset, 2

vertical-pasteboard%, [7](#)  
'vscroll, [21](#)

wheel on mouse, [3](#)  
wheel-step, [2](#)  
with-border?, [3](#)  
write-animated-gif, [14](#)  
write-gif, [14](#)