\ copyright © 2006, 2017 the osmosian order (4700)

To add a button to the dialog given a string:

If the string is blank, void the button; exit.

If the dialog is nil, exit.

Create the button.

Append the button to the dialog's buttons.

Put the string into the button's name.

To add a color and a handler to a menu:

If the menu is nil, exit.

Create a command.

Append the command to the menu's commands.

Put the swatch width into the command's width.

Put the handler into the command's handler.

Put the color into the command's color.

To add a name and a handler to a menu:

If the menu is nil, exit.

Create a command.

Append the command to the menu's commands.

Put the command height into the command's width.

Put the name into the command's name.

Put the handler into the command's handler.

Put the white color into the command's color.

Put "left" into the command's alignment.

To add a name and a handler to a submenu:

Add the name and "" and the handler to the submenu.

To add a name and a shortcut and a handler to a submenu:

If the submenu is nil, exit.

Create a command.

Append the command to the submenu's commands.

Put the submenu width into the command's width.

Put the name into the command's name.

Put the shortcut into the command's shortcut.

Put the handler into the command's handler.

Put the white color into the command's color.

Put "left" into the command's alignment.

To add swatches to a submenu given a handler:

If the submenu is nil, exit.

Add the clear color and the handler to the submenu.

Put the palette width divided by 3 into the submenu's commands' last's width.

Put "clear" into the submenu's commands' last's name.

Put "center" into the submenu's commands' last's alignment.

Add the white color and the handler to the submenu.

Put the palette width divided by 3 into the submenu's commands' last's width.

Add the black color and the handler to the submenu.

Put the palette width divided by 3 into the submenu's commands' last's width.

Add swatches to the submenu given the handler and the gray color.

Add swatches to the submenu given the handler and the red color.

Add swatches to the submenu given the handler and the orange color.

Add swatches to the submenu given the handler and the yellow color.

Add swatches to the submenu given the handler and the lime color.

Add swatches to the submenu given the handler and the green color.

Add swatches to the submenu given the handler and the teal color.

Add swatches to the submenu given the handler and the cyan color.

Add swatches to the submenu given the handler and the sky color.

Add swatches to the submenu given the handler and the blue color.

Add swatches to the submenu given the handler and the purple color.

Add swatches to the submenu given the handler and the magenta color.

Add swatches to the submenu given the handler and the violet color.

To add swatches to a submenu given a handler and a color:

If the submenu is nil, exit.

Privatize the color.

Put 875 into the color's lightness.

Loop.

Add the color and the handler to the submenu.

Subtract 125 from the color's lightness.

If the color's lightness is less than 125, exit.

Repeat.

To adjust the dialog:

If the dialog is nil, exit.

Adjust the dialog (buttons).

Adjust the dialog (roundy box).

Adjust the dialog (prompt).

Adjust the dialog (text).

To adjust the dialog (buttons):

Put the bar's display and 1/12 inch into a roundy box.

Put the roundy box's right minus the button width into the roundy box's left.

Loop.

Get a button from the dialog's buttons (backwards).

If the button is nil, exit.

Put the roundy box into the button's roundy box.

Move the roundy box left the button width.

Repeat.

To adjust the dialog (prompt):

Put the dialog's roundy box's left plus 1/8 inch into the dialog's prompt box's left.

Put the dialog's roundy box's top plus 1/24 inch into the dialog's prompt box's top.

Get a width given the dialog's prompt and the default font.

Put the dialog's prompt box's left plus the width plus 1/12 inch into the dialog's prompt box's right.

Put the dialog's roundy box's bottom minus 1/24 inch into the dialog's prompt box's bottom.

To adjust the dialog (roundy box):

Put the bar's display and 1/12 inch into the dialog's roundy box.

If the dialog's buttons are empty, exit.

Put the dialog's first button's left into the dialog's roundy box's right.

To adjust the dialog (text):

If the dialog's text is nil, exit.

Set the dialog's text's horizontal scroll flag.

Select every byte in the dialog's text.

Put the dialog's pen into the dialog's text's pen.

Put the dialog's prompt box's right into a box's left.

Put the dialog's prompt box's top into the box's top.

Put the dialog's roundy box's right minus 1/8 inch into the box's right.

Put the dialog's prompt box's bottom into the box's bottom.

Change the dialog's text given the box.

To adjust a menu given a spot and a width:

If the menu is nil, exit.

Privatize the spot.

Put the spot's x plus the width into a right coord.

Loop.

Get a command from the menu's commands.

If the command is nil, break.

If the spot's x plus the command's width is greater than the right, put the original spot's x into the spot's x; add the command height to the spot's y.

Put the spot and the spot into the command's roundy box.

Add the command's width and the command height to the command's roundy box's right-bottom.

Add the command's width to the spot's x.

Repeat.

To adjust some tabs:

If the tabs' count is 0, exit.

Put the switch's box's width divided by the tabs' count into a width.

Round the width to the tpp.

Loop.

Get a tab from the tabs.

If the tab is nil, exit.

Put the switch's box into the tab's box.

Add 1/4 inch to the tab's top.

If the tab is not the tabs' first, put the tab's previous' right into the tab's left.

If the tab is not the tabs' last, put the tab's left plus the width into the tab's right.

Repeat.

An alternative is a string.

The answer is a string.

To ask a question:

Ask the question with "".

To ask a question with choices a string and another string:

Ask the question with choices the string and the other string and "".

To ask a question with choices a string and another string and a third string:

Create the dialog.

Put "choice" into the dialog's kind.

Put the question into the dialog's prompt.

Add a button to the dialog given the string.

Add another button to the dialog given the other string.

Add a third button to the dialog given the third string.

Handle the dialog.

Destroy the dialog.

To ask a question with a default number:

Convert the default to a string.

Ask the question with the string.

To ask a question with a default ratio:

Convert the default to a string.

Ask the question with the string.

To ask a question with a default string:

Create the dialog.

Put "input" into the dialog's kind.

Put the question into the dialog's prompt.

Add a button to the dialog given "M'Kay".

Add another button to the dialog given "Cancel".

Create the dialog's text.

Put the default into the dialog's text.

Handle the dialog.

Destroy the dialog.

The bar has a height, a box, a display box and a mode.

A button is a thing with a roundy box and a name.

The button width is 2/3 inch.

To cancel the dialog:

If the dialog is nil, exit.

If the dialog's text is nil, put "cancel" into the answer.

If the dialog's text is not nil, put "" into the answer.

Relinquish control.

The command height is a height equal to 1/3 inch.

A command is a thing with

a roundy box,

a width,

a color, a font, an alignment,

a name, a shortcut, and a handler.

To create a button:

Allocate memory for the button.

To create a command:

Allocate memory for the command.

Put the clear color into the command's color.

Put the default font into the command's font.

Put 1/12 inch into the command's roundy box's radius.

Put "left" into the command's alignment.

To create the dialog:

Allocate memory for the dialog.

Put the black color into the dialog's pen.

To create a menu:

Allocate memory for the menu.

Put the big-a byte into a byte.

Loop.

If the byte is greater than the big-z byte, break.

Put the byte into a string.

Add the string and the string to the menu.

Put "center" into the menu's commands' last's alignment.

Add 1 to the byte.

Repeat.

Add "\*" and "\*" to the menu. \ special menu

Put "center" into the menu's commands' last's alignment. \ special menu

Put the bar's display's left-top into a spot. \ special menu

\ adjust the menu given the spot and 26 times the swatch width. \ no special menu

Adjust the menu given the spot and 27 times the swatch width. \ special menu

To create a submenu:

Allocate memory for the submenu.

To create a tab:

Allocate memory for the tab.

The current menu is a menu.

The current process is a process.

The current tab is a tab.

To decide if an event is ctrl-tab:

If the event is nil, say no.

If the event's kind is not "key down", say no.

If the event's key is not the tab key, say no.

If the event's ctrl flag is not set, say no.

Say yes.

To decide if a node should be read-only:

If the node is nil, say no.

If the node's kind is not "file", say no.

Loop.

Get a tab from the tabs.

If the tab is nil, say no.

If the tab's node is nil, repeat.

If the tab's node is the node, repeat.

If the tab's node's kind is not "file", repeat.

If the tab's node's path is the node's path, say yes.

Repeat.

To decide if a spot is in the bar:

If the spot is in the bar's box, say yes.

Say no.

To decide if a spot is in a button:

If the button is nil, say no.

If the spot is in the button's roundy box, say yes.

Say no.

To decide if a spot is in a command:

If the command is nil, say no.

If the spot is in the command's roundy box, say yes.

Say no.

To decide if a spot is in a menu:

If the menu is nil, say no.

Find a command given the menu and the spot.

If the command is nil, say no.

Say yes.

To decide if a spot is in the pad:

If the spot is in the pad's box, say yes.

Say no.

To decide if a spot is in the status:

If the bar's mode is not "normal", say no.

If the spot is in the status' roundy box, say yes.

Say no.

To decide if a spot is in the switch:

Find a tab given the spot.

If the tab is nil, say no.

Say yes.

To decide if a spot is in a tab:

If the tab is nil, say no.

If the spot is in the tab's box, say yes.

Say no.

To decide if a tab is free:

If the tab is nil, say no.

If the tab's node is nil, say no.

If the tab's node's kind is not "root", say no.

Say yes.

To decide if the user is sure:

Ask "Are you quite sure?" with choices "Yes" and "No".

If the answer is "yes", say yes.

Say no.

To decide if the user is willing to close a node:

If the node is nil, say yes.

If the node is not modified, say yes.

Ask "Save?" with choices "Yes" and "No" and "Cancel".

If the answer is "cancel", say no.

If the answer is "no", say yes.

Save the node.

If the i/o error is not blank, show error the i/o error; say no.

Say yes.

The desktop color is a color.

To destroy a tab:

If the tab is nil, exit.

Destroy the tab's node's root node.

Deallocate the tab.

The dialog is a thing with

a kind [choice, error, input],

a roundy box,

a prompt box, a prompt,

a pen color,

a text,

some buttons.

To draw the bar:

Unmask everything.

Draw the bar's box with the black color and the black color.

Draw the bar's box with the desktop color and the desktop color and 1/8 inch (rounding the top).

If the bar's mode is "dialog", draw the dialog; exit.

Draw the menu.

Draw the status.

To draw a box with a border color and a fill color and a radius (rounding the bottom):

Put the box and the radius into a roundy box.

Draw the roundy box with the border and the fill.

Privatize the box.

Put the box's height divided by 2 into a height.

Put the box's top plus the height into the box's bottom.

Draw the box with the border and the clear color.

Add the tpp to the box's bottom.

Draw the box with the clear color and the fill.

To draw a box with a border color and a fill color and a radius (rounding the top):

Put the box and the radius into a roundy box.

Draw the roundy box with the border and the fill.

Privatize the box.

Put the box's height divided by 2 into a height.

Put the box's bottom minus the height into the box's top.

Draw the box with the border and the clear color.

Subtract the tpp from the box's top.

Draw the box with the clear color and the fill.

To draw a button:

If the button is nil, exit.

Unmask everything.

Draw the button's roundy box with the black color and the white color.

Mask outside the button's roundy box.

Draw the button's name in the button's roundy box with the black color and the default font and "center".

To draw some buttons:

Get a button from the buttons.

If the button is nil, exit.

Draw the button.

Repeat.

To draw a command:

If the command is nil, exit.

Unmask everything.

Put the command's color into a color.

If the color is clear, put the white color into the color.

Draw the command's roundy box with the black color and the color.

Put the command's roundy box into a box.

Adjust the box given 1/8 inch and 0 and -1/8 inch and 0.

Mask outside the command's roundy box. \ don't mask to the box, menu letters are too big to fit

Draw the command's name in the box with the black color and the command's font and the command's alignment.

Draw the command's shortcut at the right of the box with the black color and the command's font.

To draw some commands:

Get a command from the commands.

If the command is nil, exit.

Draw the command.

Repeat.

To draw the desktop:

Draw the bar.

Draw the switch.

Draw the pad.

Draw the current menu.

To draw the dialog:

If the dialog is nil, exit.

Unmask everything.

Draw the dialog's roundy box with the black color and the white color.

Mask outside the dialog's roundy box.

Draw the dialog's prompt in the dialog's prompt box with the dialog's pen and the default font and "left".

Draw the dialog's text (editing).

Draw the dialog's buttons.

To draw a menu:

If the menu is nil, exit.

Draw the menu's commands.

To draw the pad:

Mask only outside the pad's box.

Draw the pad's box with the desktop color and the desktop color.

If the current tab's mode is "finder", draw the current tab's node; exit.

If the current tab's mode is "editor", draw the current tab's node's source; exit.

If the current tab's mode is "docster", draw the current tab's node's document; exit.

If the current tab's mode is "writer", draw the current tab's node's page; exit.

To draw the status:

Unmask everything.

Draw the status' roundy box with the desktop color and the desktop color.

Draw the status' roundy box with the black color and the white color.

Put the status' roundy box into a box.

Adjust the box given 1/8 inch and 0 and -1/8 inch and 0.

Mask outside the box.

Draw the status' string in the box with the black color and the default font and "right".

To draw the switch:

Mask only outside the switch's box.

Draw the switch's box with the black color and the black color.

Put the switch's box into a box.

Subtract 1/3 inch from the box's bottom.

Draw the box with the desktop color and the desktop color.

Draw the tabs.

To draw a tab:

If the tab is nil, exit.

Mask only outside the tab's box.

Draw the tab's box with the desktop color and the desktop color and 1/8 inch (rounding the bottom).

If the tab's previous is not nil, draw the tab's box's left line with the black color.

If the tab is not the current tab, draw the tab's box's top line with the black color.

Draw the tab's name in the center of the tab's box with the black color and the default font.

To draw some tabs:

Get a tab from the tabs.

If the tab is nil, exit.

Draw the tab.

Repeat.

To fill some tabs:

Get a tab from the tabs.

If the tab is nil, exit.

Put "finder" into the tab's mode.

Create the tab's node.

Open the tab's node.

Repeat.

To finalize:

Destroy the tabs.

Finalize the compiler.

Finalize the writer.

Finalize the editor.

Finalize the finder.

Finalize the desktop.

To finalize the bar:

Finalize the status.

Destroy the menu.

To finalize the desktop:

Finalize the switch.

Finalize the pad.

Finalize the bar.

To finalize the pad:

To finalize the status:

To finalize the switch:

To find a button given a key:

Void the button.

If the dialog is nil, exit.

Loop.

Get the button from the dialog's buttons.

If the button is nil, exit.

If the button's name is blank, repeat.

Put the button's name's first's target into a byte.

Uppercase the byte.

If the byte is the key, exit.

Repeat.

To find a button given a spot:

Void the button.

If the dialog is nil, exit.

Loop.

Get the button from the dialog's buttons.

If the button is nil, exit.

If the spot is in the button, exit.

Repeat.

To find a command given a menu and a spot:

Void the command.

If the menu is nil, exit.

Loop.

Get the command from the menu's commands.

If the command is nil, exit.

If the spot is in the command, exit.

Repeat.

To find a directory to compile:

Clear the directory.

If the current tab's node is nil, exit.

If the current tab's node's source is nil, exit.

Put the current tab's node's directory into the directory.

To find a free tab starting with a tab:

Privatize the tab.

Loop.

If the tab is nil, break.

If the tab is free, break.

Put the tab's next into the tab.

Repeat.

Put the tab into the free tab.

The find mode is a string [find, check spelling, check spelling - entire document, check spelling - current page only].

To find a tab given a spot:

Void the tab.

Loop.

Get the tab from the tabs.

If the tab is nil, exit.

If the spot is in the tab, exit.

Repeat.

To format a string to fit in a box given a font:

Put the box's width minus 180 into a width.

Loop.

If the string's length is less than 2, exit.

Get another width given the string and the memory canvas and the font.

If the other width is less than the width, exit.

Remove the last byte from the string.

Remove the last byte from the string.

Append "…" to the string.

Repeat.

A free tab is a tab.

To handle align given an alignment:

If the current tab's mode is "writer", handle align given the current tab's node's page and the alignment; exit.

Cluck.

To handle border and fill given a color: \ special menu

If the current tab's mode is "writer", handle border and fill given the current tab's node's page and the color; exit. \ special menu

Cluck. \ special menu

To handle border given a color:

If the current tab's mode is "writer", handle border given the current tab's node's page and the color; exit.

Cluck.

To handle bring to front:

If the current tab's mode is "writer", handle bring to front given the current tab's node's page; exit.

Cluck.

To handle capitalize:

If the current tab's mode is "editor", handle capitalize given the current tab's node's source; exit.

If the current tab's mode is "writer", handle capitalize given the current tab's node's page; exit.

Cluck.

To handle check spelling:

If the current tab's mode is "editor", handle check spelling given the current tab's node's source; exit.

Cluck.

To handle check spelling (current page only):

If the current tab's mode is "writer", handle check spelling given the current tab's node's page (current page only); exit.

Cluck.

To handle check spelling (entire document):

If the current tab's mode is "writer", handle check spelling given the current tab's node's page (entire document); exit.

Cluck.

To handle close:

If the current tab's mode is "writer", handle close (writer); exit.

Handle close (other).

To handle close (other):

If the current tab's node is root-level, cluck; deselect the nodettes in the current tab's node; show the desktop with status; exit.

If the user is not willing to close the current tab's node, exit.

Show status "Closing...".

Close the current tab's node.

Switch the current tab to the current tab's node's parent node.

Show the desktop with status.

Refresh the cursor.

To handle close (writer):

Handle close without drawing (writer).

Show the desktop with status.

Refresh the cursor.

To handle close all:

Show status "Closing...".

Get a tab from the tabs (backwards).

If the tab is nil, break.

Switch to the tab.

If the tab's node is modified, show the desktop.

If the user is not willing to close the tab's node, break.

Collapse the tab's node to root-level.

Deselect the nodettes in the tab's node.

Switch the tab to the tab's node.

Repeat.

Show the desktop with status.

To handle close without drawing (writer):

Close the current tab's node's page.

Void the current tab's node's page.

Switch the current tab to the current tab's node.

To handle a command:

If the command is nil, exit.

\ offer it dispatchers

If the command's handler is "a", handle the command (a); exit.

If the command's handler is "alignment...", handle the command (alignment...); exit.

If the command's handler is "b", handle the command (b); exit.

If the command's handler is "border...", handle the command (border...); exit.

If the command's handler is "c", handle the command (c); exit.

If the command's handler is "check spelling...", handle the command (check spelling...); exit.

If the command's handler is "color...", handle the command (color...); exit.

If the command's handler is "d", handle the command (d); exit.

If the command's handler is "e", handle the command (e); exit.

If the command's handler is "f", handle the command (f); exit.

If the command's handler is "fill...", handle the command (fill...); exit.

If the command's handler is "font height...", handle the command (font height...); exit.

If the command's handler is "g", handle the command (g); exit.

If the command's handler is "grid...", handle the command (grid...); exit.

If the command's handler is "h", handle the command (h); exit.

If the command's handler is "i", handle the command (i); exit.

If the command's handler is "j", handle the command (j); exit.

If the command's handler is "k", handle the command (k); exit.

If the command's handler is "l", handle the command (l); exit.

If the command's handler is "label...", handle the command (label...); exit.

If the command's handler is "m", handle the command (m); exit.

If the command's handler is "n", handle the command (n); exit.

If the command's handler is "o", handle the command (o); exit.

If the command's handler is "p", handle the command (p); exit.

If the command's handler is "pen...", handle the command (pen...); exit.

If the command's handler is "q", handle the command (q); exit.

If the command's handler is "r", handle the command (r); exit.

If the command's handler is "s", handle the command (s); exit.

If the command's handler is "sort...", handle the command (sort...); exit.

If the command's handler is "t", handle the command (t); exit.

If the command's handler is "u", handle the command (u); exit.

If the command's handler is "v", handle the command (v); exit.

If the command's handler is "w", handle the command (w); exit.

If the command's handler is "x", handle the command (x); exit.

If the command's handler is "y", handle the command (y); exit.

If the command's handler is "z", handle the command (z); exit.

If the command's handler is "\*", handle the command (\*); exit. \ special menu

\ do it dispatchers

Show the desktop.

If the command's handler is "handle bring to front", handle bring to front; exit.

If the command's handler is "handle align given center", handle align given "center"; exit.

If the command's handler is "handle align given left", handle align given "left"; exit.

If the command's handler is "handle align given right", handle align given "right"; exit.

If the command's handler is "handle border", handle border given the command's color; exit.

If the command's handler is "handle border and fill", handle border and fill given the command's color; exit. \ special menu

If the command's handler is "handle fill", handle fill given the command's color; exit.

If the command's handler is "handle font height given 1 inch", handle font height given 1 inch; exit.

If the command's handler is "handle font height given 1/2 inch", handle font height given 1/2 inch; exit.

If the command's handler is "handle font height given 1/3 inch", handle font height given 1/3 inch; exit.

If the command's handler is "handle font height given 1/4 inch", handle font height given 1/4 inch; exit.

If the command's handler is "handle font height given 1/5 inch", handle font height given 1/5 inch; exit.

If the command's handler is "handle font height given 1/6 inch", handle font height given 1/6 inch; exit.

If the command's handler is "handle font height given 1/8 inch", handle font height given 1/8 inch; exit.

If the command's handler is "handle font height other", handle font height other; exit.

If the command's handler is "handle font name", handle font name; exit.

If the command's handler is "handle radius", handle radius; exit.

If the command's handler is "handle pen", handle pen given the command's color; exit.

If the command's handler is "handle capitalize", handle capitalize; exit.

If the command's handler is "handle check spelling", handle check spelling; exit.

If the command's handler is "handle check spelling (entire document)", handle check spelling (entire document); exit.

If the command's handler is "handle check spelling (current page only)", handle check spelling (current page only); exit.

If the command's handler is "handle close", handle close; exit.

If the command's handler is "handle close all", handle close all; exit.

If the command's handler is "handle comment", handle comment; exit.

If the command's handler is "handle compile", handle compile; exit.

If the command's handler is "handle copy", handle copy; exit.

If the command's handler is "handle cut", handle cut; exit.

If the command's handler is "handle decrypt", handle decrypt; exit.

If the command's handler is "handle duplicate", handle duplicate; exit.

If the command's handler is "handle enlarge", handle enlarge; exit.

If the command's handler is "handle encrypt", handle encrypt; exit.

If the command's handler is "handle extract globals", handle extract globals; exit.

If the command's handler is "handle extract routine headers", handle extract routine headers; exit.

If the command's handler is "handle extract types", handle extract types; exit.

If the command's handler is "handle extract nibble literal", handle extract nibble literal; exit.

If the command's handler is "handle find", handle find; exit.

If the command's handler is "handle find next", handle find next; exit.

If the command's handler is "handle flip", handle flip; exit.

If the command's handler is "handle get size", handle get size; exit.

If the command's handler is "handle grayscale", handle grayscale given yes; exit.

If the command's handler is "handle group", handle group; exit.

If the command's handler is "handle grid given no", handle grid given no; exit.

If the command's handler is "handle grid given yes", handle grid given yes; exit.

If the command's handler is "handle indent", handle indent; exit.

If the command's handler is "handle label given title", handle label given "title"; exit.

If the command's handler is "handle label given page number", handle label given "page number"; exit.

If the command's handler is "handle label given page number (even)", handle label given "page number (even)"; exit.

If the command's handler is "handle label given page number (odd)", handle label given "page number (odd)"; exit.

If the command's handler is "handle list", handle list; exit.

If the command's handler is "handle lowercase", handle lowercase; exit.

If the command's handler is "handle minimize", handle minimize; exit.

If the command's handler is "handle mirror", handle mirror; exit.

If the command's handler is "handle new directory", handle new directory; exit.

If the command's handler is "handle new document", handle new document; exit.

If the command's handler is "handle new ellipse", handle new shape given "ellipse"; exit.

If the command's handler is "handle new page", handle new page; exit.

If the command's handler is "handle new picture", handle new shape given "picture"; exit.

If the command's handler is "handle new polygon", handle new shape given "polygon"; exit.

If the command's handler is "handle new rectangle", handle new shape given "rectangle"; exit.

If the command's handler is "handle new text", handle new shape given "text"; exit.

If the command's handler is "handle new text file", handle new text file; exit.

If the command's handler is "handle open", handle open; exit.

If the command's handler is "handle open as dump", handle open as dump; exit.

If the command's handler is "handle open as text", handle open as text; exit.

If the command's handler is "handle outdent", handle outdent; exit.

If the command's handler is "handle paste", handle paste; exit.

If the command's handler is "handle print", handle print; exit.

If the command's handler is "handle quit", handle quit; exit.

If the command's handler is "handle redo", handle redo; exit.

If the command's handler is "handle reduce", handle reduce; exit.

If the command's handler is "handle rename", handle rename; exit.

If the command's handler is "handle renumber pages", handle renumber pages; exit.

If the command's handler is "handle reverse", handle reverse; exit.

If the command's handler is "handle rotate page", handle rotate page; exit.

If the command's handler is "handle rotate shape", handle rotate shape; exit.

If the command's handler is "handle run", handle run; exit.

If the command's handler is "handle save", handle save; exit.

If the command's handler is "handle save all", handle save all; exit.

If the command's handler is "handle save as pdf", handle save as pdf; exit.

If the command's handler is "handle select all", handle select all; exit.

If the command's handler is "handle send to back", handle send to back; exit.

If the command's handler is "handle measure", handle measure; exit.

If the command's handler is "handle smooth", handle smooth; exit.

If the command's handler is "handle yank given 1/8 inch", handle yank given 1/8 inch; exit.

If the command's handler is "handle yank given 1/12 inch", handle yank given 1/12 inch; exit.

If the command's handler is "handle yank given 1/16 inch", handle yank given 1/16 inch; exit.

If the command's handler is "handle yank given 1/24 inch", handle yank given 1/24 inch; exit.

If the command's handler is "handle yank given 1/32 inch", handle yank given 1/32 inch; exit.

If the command's handler is "handle yank given 1/48 inch", handle yank given 1/48 inch; exit.

If the command's handler is "handle yank given the tpp", handle yank given the tpp; exit.

If the command's handler is "handle sort definitions", handle sort definitions; exit.

If the command's handler is "handle sort selected lines", handle sort selected lines; exit.

If the command's handler is "handle sort nodettes by extension", handle sort nodettes given "extension"; exit.

If the command's handler is "handle sort nodettes by kind", handle sort nodettes given "kind"; exit.

If the command's handler is "handle sort nodettes by name", handle sort nodettes given "name"; exit.

If the command's handler is "handle stop", handle stop; exit.

If the command's handler is "handle uncomment", handle uncomment; exit.

If the command's handler is "handle undo", handle undo; exit.

If the command's handler is "handle ungrayscale", handle grayscale given no; exit.

If the command's handler is "handle ungroup", handle ungroup; exit.

If the command's handler is "handle unlabel", handle unlabel; exit.

If the command's handler is "handle uppercase", handle uppercase; exit.

If the command's handler is "handle version", handle version; exit.

\ not properly dispatched

Debug "Internal error 763 - command not properly dispatched".

To handle a command (\*): \ special menu

Create a submenu. \ special menu

Add swatches to the submenu given "handle border and fill". \ special menu

Handle the command given the submenu and the palette width. \ special menu

To handle a command (a):

Create a submenu.

If the current tab's mode is "writer", add "Align Left" and "handle align given left" to the submenu.

If the current tab's mode is "writer", add "Align Center" and "handle align given center" to the submenu.

If the current tab's mode is "writer", add "Align Right" and "handle align given right" to the submenu.

Handle the command given the submenu.

To handle a command (alignment...):

Create a submenu.

Add "Align Left" and "handle align given left" to the submenu.

Add "Align Center" and "handle align given center" to the submenu.

Add "Align Right" and "handle align given right" to the submenu.

Handle the command given the submenu.

To handle a command (b):

Create a submenu.

If the current tab's mode is "writer", add "Border..." and "border..." to the submenu.

If the current tab's mode is "writer", add "Bring to Front" and "handle bring to front" to the submenu.

Handle the command given the submenu.

To handle a command (border...):

Create a submenu.

Add swatches to the submenu given "handle border".

Handle the command given the submenu and the palette width.

To handle a command (c):

Create a submenu.

If the current tab's mode is "editor", add "Capitalize" and "handle capitalize" to the submenu.

If the current tab's mode is "writer", add "Capitalize" and "handle capitalize" to the submenu.

If the current tab's mode is "editor", add "Check Spelling" and "handle check spelling" to the submenu.

If the current tab's mode is "writer", add "Check Spelling..." and "check spelling..." to the submenu.

Add "Close" and "W" and "handle close" to the submenu.

Add "Close All" and "handle close all" to the submenu.

If the current tab's mode is "writer", add "Color..." and "color..." to the submenu.

If the current tab's mode is "editor", add "Comment" and "T" and "handle comment" to the submenu.

If the current tab's mode is "editor", add "Compile" and "M" and "handle compile" to the submenu.

Add "Copy" and "C" and "handle copy" to the submenu.

Add "Cut" and "X" and "handle cut" to the submenu.

Handle the command given the submenu.

To handle a command (check spelling...):

Create a submenu.

Add "Current Page Only" and "handle check spelling (current page only)" to the submenu.

Add "Entire Document" and "handle check spelling (entire document)" to the submenu.

Handle the command given the submenu.

To handle a command (color...):

Create a submenu.

Add "Border..." and "border..." to the submenu.

Add "Fill..." and "fill..." to the submenu.

Add "Pen..." and "pen..." to the submenu.

Handle the command given the submenu.

To handle a command (d):

Create a submenu.

If the current tab's mode is "editor", add "Decrypt..." and "handle decrypt" to the submenu.

If the current tab's mode is "finder", add "Duplicate" and "D" and "handle duplicate" to the submenu.

If the current tab's mode is "docster", add "Duplicate" and "D" and "handle duplicate" to the submenu.

If the current tab's mode is "writer", add "Duplicate" and "D" and "handle duplicate" to the submenu.

Handle the command given the submenu.

To handle a command (e):

Create a submenu.

If the current tab's mode is "writer", add "Enlarge" and "E" and "handle enlarge" to the submenu.

If the current tab's mode is "editor", add "Encrypt..." and "handle encrypt" to the submenu.

If the current tab's mode is "editor", add "Extract Types" and "handle extract types" to the submenu.

If the current tab's mode is "editor", add "Extract Globals" and "handle extract globals" to the submenu.

If the current tab's mode is "editor", add "Extract Routine Headers" and "handle extract routine headers" to the submenu.

If the current tab's mode is "finder", add "Extract Nibble Literal" and "handle extract nibble literal" to the submenu.

Handle the command given the submenu.

To handle a command (f):

Create a submenu.

If the current tab's mode is "editor", add "Find" and "F" and "handle find" to the submenu.

If the current tab's mode is "writer", add "Find" and "F" and "handle find" to the submenu.

If the current tab's mode is "editor", add "Find Next" and "N" and "handle find next" to the submenu.

If the current tab's mode is "writer", add "Find Next" and "N" and "handle find next" to the submenu.

If the current tab's mode is "writer", add "Fill..." and "fill..." to the submenu.

If the current tab's mode is "writer", add "Flip" and "handle flip" to the submenu.

If the current tab's mode is "editor", add "Font Height..." and "font height..." to the submenu.

If the current tab's mode is "editor", add "Font Name..." and "handle font name" to the submenu.

If the current tab's mode is "writer", add "Font Height..." and "font height..." to the submenu.

If the current tab's mode is "writer", add "Font Name..." and "handle font name" to the submenu.

Handle the command given the submenu.

To handle a command (fill...):

Create a submenu.

Add swatches to the submenu given "handle fill".

Handle the command given the submenu and the palette width.

To handle a command (font height...):

Create a submenu.

Add "1 Inch" and "handle font height given 1 inch" to the submenu.

Add "1/2 Inch" and "handle font height given 1/2 inch" to the submenu.

Add "1/3 Inch" and "handle font height given 1/3 inch" to the submenu.

Add "1/4 Inch" and "handle font height given 1/4 inch" to the submenu.

Add "1/5 Inch" and "handle font height given 1/5 inch" to the submenu.

Add "1/6 Inch" and "handle font height given 1/6 inch" to the submenu.

Add "1/8 Inch" and "handle font height given 1/8 inch" to the submenu.

Add "Other..." and "handle font height other" to the submenu.

Handle the command given the submenu.

To handle a command (g):

Create a submenu.

If the current tab's mode is "finder", add "Get Size" and "handle get size" to the submenu.

If the current tab's mode is "writer", add "Grayscale" and "handle grayscale" to the submenu.

If the current tab's mode is "writer", add "Grid..." and "grid..." to the submenu.

If the current tab's mode is "writer", add "Group" and "G" and "handle group" to the submenu.

Handle the command given the submenu.

To handle a command (grid...):

Create a submenu.

Add "Hide" and "handle grid given no" to the submenu.

Add "Show" and "handle grid given yes" to the submenu.

Handle the command given the submenu.

To handle a command (h):

Create a submenu.

If the current tab's mode is "writer", add "Hide Grid" and "handle grid given no" to the submenu.

Handle the command given the submenu.

To handle a command (i):

Create a submenu.

If the current tab's mode is "editor", add "Indent" and "I" and "handle indent" to the submenu.

Handle the command given the submenu.

To handle a command (j):

To handle a command (k):

To handle a command (l):

Create a submenu.

If the current tab's mode is "editor", add "List" and "L" and "handle list" to the submenu.

If the current tab's mode is "editor", add "Lowercase" and "handle lowercase" to the submenu.

If the current tab's mode is "writer", add "Label..." and "Label..." to the submenu.

If the current tab's mode is "writer", add "Lowercase" and "handle lowercase" to the submenu.

Handle the command given the submenu.

To handle a command (label...):

Create a submenu.

Add "Title" and "handle label given title" to the submenu.

Add "Page Number" and "handle label given page number" to the submenu.

Add "Page Number (Even)" and "handle label given page number (even)" to the submenu.

Add "Page Number (Odd)" and "handle label given page number (odd)" to the submenu.

Handle the command given the submenu.

To handle a command (m):

Create a submenu.

Add "Minimize" and "handle minimize" to the submenu.

If the current tab's mode is "writer", add "Measure..." and "handle measure" to the submenu.

If the current tab's mode is "writer", add "Mirror" and "handle mirror" to the submenu.

Handle the command given the submenu.

To handle a command (n):

Create a submenu.

If the current tab's mode is "finder", add "New Directory..." and "handle new directory" to the submenu.

If the current tab's mode is "finder", add "New Document..." and "handle new document" to the submenu.

If the current tab's mode is "writer", add "New Ellipse" and "handle new ellipse" to the submenu.

If the current tab's mode is "docster", add "New Page" and "handle new page" to the submenu.

If the current tab's mode is "writer", add "New Picture..." and "handle new picture" to the submenu.

If the current tab's mode is "writer", add "New Polygon" and "handle new polygon" to the submenu.

If the current tab's mode is "writer", add "New Rectangle" and "handle new rectangle" to the submenu.

If the current tab's mode is "writer", add "New Text" and "handle new text" to the submenu.

If the current tab's mode is "finder", add "New Text File..." and "handle new text file" to the submenu.

Handle the command given the submenu.

To handle a command (o):

Create a submenu.

If the current tab's mode is "finder", add "Open" and "O" and "handle open" to the submenu.

If the current tab's mode is "docster", add "Open" and "O" and "handle open" to the submenu.

If the current tab's mode is "finder", add "Open As Dump" and "handle open as dump" to the submenu.

If the current tab's mode is "finder", add "Open As Text" and "handle open as text" to the submenu.

If the current tab's mode is "editor", add "Outdent" and "O" and "handle outdent" to the submenu.

Handle the command given the submenu.

To handle a command (p):

Create a submenu.

Add "Paste" and "V" and "handle paste" to the submenu.

If the current tab's mode is "writer", add "Pen..." and "pen..." to the submenu.

If the current tab's mode is "editor", add "Print..." and "P" and "handle print" to the submenu.

If the current tab's mode is "docster", add "Print..." and "P" and "handle print" to the submenu.

If the current tab's mode is "writer", add "Print..." and "P" and "handle print" to the submenu.

Handle the command given the submenu.

To handle a command (pen...):

Create a submenu.

Add swatches to the submenu given "handle pen".

Handle the command given the submenu and the palette width.

To handle a command (q):

Create a submenu.

Add "Quit" and "Q" and "handle quit" to the submenu.

Handle the command given the submenu.

To handle a command (r):

Create a submenu.

If the current tab's mode is "writer", add "Radius..." and "handle radius" to the submenu.

If the current tab's mode is "editor", add "Redo" and "Y" and "handle redo" to the submenu.

If the current tab's mode is "writer", add "Redo" and "Y" and "handle redo" to the submenu.

If the current tab's mode is "writer", add "Reduce" and "R" and "handle reduce" to the submenu.

If the current tab's mode is "finder", add "Rename..." and "handle rename" to the submenu.

If the current tab's mode is "docster", add "Renumber Pages..." and "handle renumber pages" to the submenu.

If the current tab's mode is "editor", add "Reverse" and "handle reverse" to the submenu.

If the current tab's mode is "finder", add "Reverse" and "handle reverse" to the submenu.

If the current tab's mode is "writer", add "Rotate Page" and "handle rotate page" to the submenu.

If the current tab's mode is "writer", add "Rotate Shape" and "J" and "handle rotate shape" to the submenu.

If the current tab's mode is "finder", add "Run" and "R" and "handle run" to the submenu.

If the current tab's mode is "editor", add "Run" and "R" and "handle run" to the submenu.

Handle the command given the submenu.

To handle a command (s):

Create a submenu.

If the current tab's mode is "editor", add "Save" and "S" and "handle save" to the submenu.

If the current tab's mode is "docster", add "Save" and "S" and "handle save" to the submenu.

If the current tab's mode is "writer", add "Save" and "S" and "handle save" to the submenu.

Add "Save All" and "handle save all" to the submenu.

If the current tab's mode is "docster", add "Save as PDF..." and "handle save as pdf" to the submenu.

Add "Select All" and "A" and "handle select all" to the submenu.

If the current tab's mode is "writer", add "Send to Back" and "handle send to back" to the submenu.

If the current tab's mode is "writer", add "Show Grid" and "handle grid given yes" to the submenu.

If the current tab's mode is "writer", add "Smooth" and "M" and "handle smooth" to the submenu.

If the current tab's mode is "editor", add "Sort..." and "sort..." to the submenu.

If the current tab's mode is "finder", add "Sort..." and "sort..." to the submenu.

If the current tab's mode is "editor", add "Stop" and "handle stop" to the submenu.

If the current tab's mode is "finder", add "Stop" and "handle stop" to the submenu.

Handle the command given the submenu.

To handle a command (sort...):

Create a submenu.

If the current tab's mode is "editor", add "Definitions" and "handle sort definitions" to the submenu.

If the current tab's mode is "editor", add "Selected Lines" and "handle sort selected lines" to the submenu.

If the current tab's mode is "finder", add "By Kind" and "handle sort nodettes by kind" to the submenu.

If the current tab's mode is "finder", add "By Name" and "handle sort nodettes by name" to the submenu.

If the current tab's mode is "finder", add "By Extension" and "handle sort nodettes by extension" to the submenu.

Handle the command given the submenu.

To handle a command (t):

To handle a command (u):

Create a submenu.

If the current tab's mode is "editor", add "Uncomment" and "U" and "handle uncomment" to the submenu.

If the current tab's mode is "editor", add "Undo" and "Z" and "handle undo" to the submenu.

If the current tab's mode is "writer", add "Undo" and "Z" and "handle undo" to the submenu.

If the current tab's mode is "writer", add "Ungrayscale" and "handle ungrayscale" to the submenu.

If the current tab's mode is "writer", add "Ungroup" and "U" and "handle ungroup" to the submenu.

If the current tab's mode is "writer", add "Unlabel" and "handle unlabel" to the submenu.

If the current tab's mode is "editor", add "Uppercase" and "handle uppercase" to the submenu.

If the current tab's mode is "writer", add "Uppercase" and "handle uppercase" to the submenu.

Handle the command given the submenu.

To handle a command (v):

Create a submenu.

Add "Version" and "handle version" to the submenu.

Handle the command given the submenu.

To handle a command (w):

To handle a command (x):

To handle a command (y):

Create a submenu.

If the current tab's mode is "writer", add "Yank 1/8" and "handle yank given 1/8 inch" to the submenu.

If the current tab's mode is "writer", add "Yank 1/12" and "handle yank given 1/12 inch" to the submenu.

If the current tab's mode is "writer", add "Yank 1/16" and "handle yank given 1/16 inch" to the submenu.

If the current tab's mode is "writer", add "Yank 1/24" and "handle yank given 1/24 inch" to the submenu.

If the current tab's mode is "writer", add "Yank 1/32" and "handle yank given 1/32 inch" to the submenu.

If the current tab's mode is "writer", add "Yank 1/48" and "handle yank given 1/48 inch" to the submenu.

If the current tab's mode is "writer", add "Yank None" and "handle yank given the tpp" to the submenu.

Handle the command given the submenu.

To handle a command (z):

To handle a command given a submenu: handle the command given the submenu and the submenu width.

To handle a command given a submenu and a width:

Put the command's left and the bar's display's bottom into a spot.

Adjust the submenu given the spot and the width.

Flush all events.

Void the selected command.

Put the submenu into the current menu.

Show the desktop.

Handle events given the submenu.

Void the current menu.

Handle the selected command.

Destroy the submenu.

Show the desktop.

Refresh the cursor.

To handle comment:

If the current tab's mode is "editor", handle comment given the current tab's node's source; exit.

Cluck.

To handle compile:

If the current tab's mode is not "editor", cluck; exit.

Handle save all.

If the i/o error is not blank, exit.

Find a directory to compile.

If the directory is blank, show error "I'm not sure what to compile."; exit.

Compile the directory.

If the compiler's abort flag is set, handle compile or list or run error; exit.

Show status the compiler's timer's string then " ms".

To handle compile or list or run error:

If the compiler's abort path is blank, show error the compiler's abort message; exit.

Loop.

Get a tab from the tabs.

If the tab is nil, show error the compiler's abort message; exit.

If the compiler's abort path is not the tab's node's path, repeat.

If the tab's node's source is nil, show error the compiler's abort message; exit.

Select the compiler's abort row# given the tab's node's source's text.

Scroll the tab's node's source's text to the caret and center it.

Switch to the tab.

Show the desktop.

Show error the compiler's abort message.

To handle copy:

If the current tab's mode is "finder", handle copy given the current tab's node; exit.

If the current tab's mode is "editor", handle copy given the current tab's node's source; exit.

If the current tab's mode is "docster", handle copy given the current tab's node's document; exit.

If the current tab's mode is "writer", handle copy given the current tab's node's page; exit.

Cluck.

To handle cut:

If the current tab's mode is "finder", handle cut given the current tab's node; exit.

If the current tab's mode is "editor", handle cut given the current tab's node's source; exit.

If the current tab's mode is "docster", handle cut given the current tab's node's document; exit.

If the current tab's mode is "writer", handle cut given the current tab's node's page; exit.

Cluck.

To handle decrypt:

If the current tab's mode is "editor", handle decrypt given the current tab's node's source; exit.

Cluck.

To handle the dialog:

If the dialog is nil, exit.

Flush all events.

Adjust the dialog.

Switch the bar to "dialog".

Handle events given the dialog.

Switch the bar to "normal" without drawing.

If the answer is blank, show the desktop.

If the dialog's text is nil, show the desktop.

Flush all events.

Refresh the cursor.

To handle duplicate:

If the current tab's mode is "finder", handle duplicate given the current tab's node; exit.

If the current tab's mode is "docster", handle duplicate given the current tab's node's document; exit.

If the current tab's mode is "writer", handle duplicate given the current tab's node's page; exit.

Cluck.

To handle encrypt:

If the current tab's mode is "editor", handle encrypt given the current tab's node's source; exit.

Cluck.

To handle enlarge:

If the current tab's mode is "writer", handle enlarge given the current tab's node's page; exit.

Cluck.

To handle an event:

If the event is nil, exit.

If the event's kind is "key down", handle the event (key down); exit.

If the event's kind is "left click", handle the event (left click); exit.

If the event's kind is "left double click", handle the event (left double click); exit.

If the event's kind is "refresh", handle the event (refresh); exit.

If the event's kind is "right click", handle the event (right click); exit.

If the event's kind is "set cursor", handle the event (set cursor); exit.

To handle an event (key down - ctrl-tab):

If the event's shift flag is set, switch to the predecessor of the current tab.

If the event's shift flag is not set, switch to the successor of the current tab.

Show the desktop with status.

To handle an event (key down - shortcut):

If the event's key is the a-key, handle select all; exit.

If the event's key is the c-key, handle copy; exit.

If the event's key is the d-key, handle duplicate; exit.

If the event's key is the e-key, handle enlarge; exit.

If the event's key is the f-key, handle find; exit.

If the event's key is the g-key, handle group; exit.

If the event's key is the i-key, handle indent; exit.

If the event's key is the j-key, handle rotate shape; exit.

If the event's key is the l-key, handle list; exit.

If the event's key is the m-key and the current tab's mode is "editor", handle compile; exit.

If the event's key is the m-key and the current tab's mode is "writer", handle smooth; exit.

If the event's key is the n-key, handle find next; exit.

If the event's key is the o-key and the current tab's mode is "editor", handle outdent; exit.

If the event's key is the o-key, handle open; exit.

If the event's key is the p-key, handle print; exit.

If the event's key is the q-key, handle quit; exit.

If the event's key is the r-key and the current tab's mode is "editor", handle run; exit.

If the event's key is the r-key and the current tab's mode is "finder", handle run; exit.

If the event's key is the r-key and the current tab's mode is "writer", handle reduce; exit.

If the event's key is the s-key, handle save; exit.

If the event's key is the t-key, handle comment; exit.

If the event's key is the u-key and the current tab's mode is "editor", handle uncomment; exit.

If the event's key is the u-key and the current tab's mode is "writer", handle ungroup; exit.

If the event's key is the v-key, handle paste; exit.

If the event's key is the w-key, handle close; exit.

If the event's key is the x-key, handle cut; exit.

If the event's key is the y-key, handle redo; exit.

If the event's key is the z-key, handle undo; exit.

Cluck.

To handle an event (key down):

If the event is any shortcut, handle the event (key down - shortcut); exit.

If the event is ctrl-tab, handle the event (key down - ctrl-tab); exit.

If the current tab's mode is "finder", handle the event given the current tab's node (key down); exit.

If the current tab's mode is "editor", handle the event given the current tab's node's source (key down); exit.

If the current tab's mode is "docster", handle the event given the current tab's node's document (key down); exit.

If the current tab's mode is "writer", handle the event given the current tab's node's page (key down); exit.

To handle an event (left click - menu):

Find a command given the menu and the event's spot.

Handle the command.

To handle an event (left click - status):

Track the desktop given the event.

To handle an event (left click - switch):

Find a tab given the event's spot.

If the tab is nil, exit.

If the tab is the current tab, set a flag.

Switch to the tab.

Show the desktop with status.

Track the tab returning another flag.

If the other flag is set, exit.

If the flag is not set, exit.

Handle close.

To handle an event (left click):

If the event's spot is in the menu, handle the event (left click - menu); exit.

If the event's spot is in the status, handle the event (left click - status); exit.

If the event's spot is in the switch, handle the event (left click - switch); exit.

If the current tab's mode is "finder", handle the event given the current tab's node (left click); exit.

If the current tab's mode is "editor", handle the event given the current tab's node's source (left click); exit.

If the current tab's mode is "docster", handle the event given the current tab's node's document (left click); exit.

If the current tab's mode is "writer", handle the event given the current tab's node's page (left click); exit.

To handle an event (left double click):

If the event's spot is in the bar, exit.

If the event's spot is in the switch, exit.

If the current tab's mode is "finder", handle the event given the current tab's node (left double click); exit.

If the current tab's mode is "editor", handle the event given the current tab's node's source (left double click); exit.

If the current tab's mode is "docster", handle the event given the current tab's node's document (left double click); exit.

If the current tab's mode is "writer", handle the event given the current tab's node's page (left double click); exit.

To handle an event (refresh):

Show the desktop with status.

To handle an event (right click):

If the current tab's mode is "finder", handle the event given the current tab's node (right click); exit.

If the current tab's mode is "editor", handle the event given the current tab's node's source (right click); exit.

If the current tab's mode is "docster", handle the event given the current tab's node's document (right click); exit.

If the current tab's mode is "writer", handle the event given the current tab's node's page (right click); exit.

To handle an event (set cursor):

If the current tab's mode is "finder", handle the event given the current tab's node (set cursor); exit.

If the current tab's mode is "editor", handle the event given the current tab's node's source (set cursor); exit.

If the current tab's mode is "docster", handle the event given the current tab's node's document (set cursor); exit.

If the current tab's mode is "writer", handle the event given the current tab's node's page (set cursor); exit.

Show the arrow cursor.

To handle an event given the dialog:

If the event's kind is "key down", handle the event given the dialog (key down); exit.

If the event's kind is "left click", handle the event given the dialog (left click); exit.

If the event's kind is "left double click", handle the event given the dialog (left double click); exit.

If the event's kind is "refresh", handle the event given the dialog (refresh); exit.

If the event's kind is "right click", handle the event given the dialog (right click); exit.

If the event's kind is "set cursor", handle the event given the dialog (set cursor); exit.

To handle an event given the dialog (copy):

If nothing is selected in the dialog's text, cluck; exit.

Handle copy given the dialog's text.

To handle an event given the dialog (cut):

If nothing is selected in the dialog's text, cluck; exit.

Handle cut given the dialog's text.

Show the desktop.

To handle an event given the dialog (key down - backspace key):

Handle the event given the dialog's text (backspace key).

Show the desktop.

To handle an event given the dialog (key down - choice):

If the event's key is the enter key, put "yes" into the answer; relinquish control; exit.

If the event's key is the escape key, put "cancel" into the answer; relinquish control; exit.

Find a button given the event's key.

If the button is nil, exit.

Put the button's name into the answer.

Relinquish control.

To handle an event given the dialog (key down - delete key):

Handle the event given the dialog's text (delete key).

Show the desktop.

To handle an event given the dialog (key down - end key):

Handle the event given the dialog's text (end key).

Show the desktop.

To handle an event given the dialog (key down - enter key):

Extract the answer from the dialog's text.

Relinquish control.

To handle an event given the dialog (key down - error):

Cancel the dialog.

To handle an event given the dialog (key down - escape key):

Cancel the dialog.

To handle an event given the dialog (key down - home key):

Handle the event given the dialog's text (home key).

Show the desktop.

To handle an event given the dialog (key down - input):

If the event is any shortcut, handle the event given the dialog (key down - shortcut); exit.

If the event's byte is printable, handle the event given the dialog (key down - printable key); exit.

If the event's key is the backspace key, handle the event given the dialog (key down - backspace key); exit.

If the event's key is the delete key, handle the event given the dialog (key down - delete key); exit.

If the event's key is the end key, handle the event given the dialog (key down - end key); exit.

If the event's key is the enter key, handle the event given the dialog (key down - enter key); exit.

If the event's key is the escape key, handle the event given the dialog (key down - escape key); exit.

If the event's key is the home key, handle the event given the dialog (key down - home key); exit.

If the event's key is the left-arrow key, handle the event given the dialog (key down - left-arrow key); exit.

If the event's key is the right-arrow key, handle the event given the dialog (key down - right-arrow key); exit.

To handle an event given the dialog (key down - left-arrow key):

Handle the event given the dialog's text (left-arrow key).

Show the desktop.

To handle an event given the dialog (key down - printable key):

Handle the event given the dialog's text (printable key).

Show the desktop.

To handle an event given the dialog (key down - right-arrow key):

Handle the event given the dialog's text (right-arrow key).

Show the desktop.

To handle an event given the dialog (key down - shortcut):

If the event's key is the a-key, handle the event given the dialog (select all).

If the event's key is the c-key, handle the event given the dialog (copy).

If the event's key is the v-key, handle the event given the dialog (paste).

If the event's key is the x-key, handle the event given the dialog (cut).

If the event's key is the y-key, handle the event given the dialog (redo).

If the event's key is the z-key, handle the event given the dialog (undo).

To handle an event given the dialog (key down):

If the dialog's kind is "choice", handle the event given the dialog (key down - choice); exit.

If the dialog's kind is "error", handle the event given the dialog (key down - error); exit.

If the dialog's kind is "input", handle the event given the dialog (key down - input); exit.

To handle an event given the dialog (left click - choice):

If the event's spot is not in the bar, cancel the dialog; exit.

Find a button given the event's spot.

If the button is nil, exit.

Put the button's name into the answer.

Relinquish control.

To handle an event given the dialog (left click - error):

If the event's spot is not in the bar, cancel the dialog; exit.

Find a button given the event's spot.

If the button is nil, exit.

Put the button's name into the answer.

Relinquish control.

To handle an event given the dialog (left click - input):

If the event's spot is not in the bar, cancel the dialog; exit.

If the event's spot is in the dialog's text, track the dialog given the event; exit.

Find a button given the event's spot.

If the button is nil, exit.

If the button's name is "cancel", cancel the dialog; exit.

Extract the answer from the dialog's text.

Relinquish control.

To handle an event given the dialog (left click):

If the dialog's kind is "choice", handle the event given the dialog (left click - choice); exit.

If the dialog's kind is "error", handle the event given the dialog (left click - error); exit.

If the dialog's kind is "input", handle the event given the dialog (left click - input); exit.

To handle an event given the dialog (left double click):

If the dialog's text is nil, exit.

Handle the event given the dialog's text (left double click).

Show the desktop.

To handle an event given the dialog (paste):

If there is not text on the windows clipboard, cluck; exit.

Handle paste given the dialog's text.

Show the desktop.

To handle an event given the dialog (redo):

If the dialog's text cannot be redone, cluck; exit.

Handle redo given the dialog's text.

Show the desktop.

To handle an event given the dialog (refresh):

Show the desktop.

To handle an event given the dialog (right click):

If the event's spot is in the bar, exit.

Cancel the dialog.

To handle an event given the dialog (select all):

Handle select all given the dialog's text.

Show the desktop.

To handle an event given the dialog (set cursor):

If the event's spot is in the dialog's text, show the i-beam cursor; exit.

Show the arrow cursor.

To handle an event given the dialog (undo):

If the dialog's text cannot be undone, cluck; exit.

Handle undo given the dialog's text.

Show the desktop.

To handle an event given a submenu:

If the event's kind is "key down", handle the event given the submenu (key down); exit.

If the event's kind is "left click", handle the event given the submenu (left click); exit.

If the event's kind is "refresh", handle the event given the submenu (refresh); exit.

If the event's kind is "set cursor", handle the event given the submenu (set cursor); exit.

To handle an event given a submenu (key down):

If the event's key is the escape key, relinquish control.

To handle an event given a submenu (left click - menu):

Find a command given the submenu and the event's spot.

If the command is nil, exit.

Put the command into the selected command.

Relinquish control.

To handle an event given a submenu (left click):

If the event's spot is in the submenu, handle the event given the submenu (left click - menu); exit.

Relinquish control.

If the event's spot is in the menu, reque the event.

To handle an event given a submenu (refresh):

Show the desktop.

To handle an event given a submenu (set cursor):

Show the arrow cursor.

To handle events:

Deque an event.

If the event is nil, exit.

Handle the event.

Repeat.

To handle events given the dialog:

Deque an event.

If the event is nil, exit.

Handle the event given the dialog.

Repeat.

To handle events given a submenu:

Deque an event.

If the event is nil, exit.

Handle the event given the submenu.

Repeat.

To handle extract globals:

If the current tab's mode is "editor", handle extract globals given the current tab's node's source; exit.

Cluck.

To handle extract nibble literal:

If the current tab's mode is "finder", handle extract nibble literal given the current tab's node; exit.

Cluck.

To handle extract routine headers:

If the current tab's mode is "editor", handle extract routine headers given the current tab's node's source; exit.

Cluck.

To handle extract types:

If the current tab's mode is "editor", handle extract types given the current tab's node's source; exit.

Cluck.

To handle fill given a color:

If the current tab's mode is "writer", handle fill given the current tab's node's page and the color; exit.

Cluck.

To handle find:

If the current tab's mode is "editor", handle find given the current tab's node's source; exit.

If the current tab's mode is "writer", handle find given the current tab's node's page; exit.

Cluck.

To handle find next:

If the current tab's mode is "editor", handle find next given the current tab's node's source; exit.

If the current tab's mode is "writer", handle find next given the current tab's node's page; exit.

Cluck.

To handle flip:

If the current tab's mode is "writer", handle flip given the current tab's node's page; exit.

Cluck.

To handle font height given a font height:

If the current tab's mode is "editor", handle font height given the current tab's node's source and the font height; exit.

If the current tab's mode is "writer", handle font height given the current tab's node's page and the font height; exit.

Cluck.

To handle font height other:

If the current tab's mode is "editor", handle font height other given the current tab's node's source; exit.

If the current tab's mode is "writer", handle font height other given the current tab's node's page; exit.

To handle font name:

If the current tab's mode is "editor", handle font name given the current tab's node's source; exit.

If the current tab's mode is "writer", handle font name given the current tab's node's page; exit.

To handle get size:

If the current tab's mode is "finder", handle get size given the current tab's node; exit.

Cluck.

To handle grayscale given a flag:

If the current tab's mode is "writer", handle grayscale given the current tab's node's page and the flag; exit.

Cluck.

To handle grid given a flag:

If the current tab's mode is "writer", handle grid given the current tab's node's page and the flag; exit.

Cluck.

To handle group:

If the current tab's mode is "writer", handle group given the current tab's node's page; exit.

Cluck.

To handle indent:

If the current tab's mode is "editor", handle indent given the current tab's node's source; exit.

Cluck.

To handle label given a label:

If the current tab's mode is "writer", handle label given the current tab's node's page and the label; exit.

Cluck.

To handle list:

If the current tab's mode is not "editor", cluck; exit.

Handle save all.

If the i/o error is not blank, exit.

Find a directory to compile.

If the directory is blank, show error "I'm not sure what to compile and list"; exit.

Compile the directory.

Show status "Listing...".

List.

If the compiler's abort flag is set, handle compile or list or run error; exit.

Show status the compiler's timer's string then " ms".

To handle lowercase:

If the current tab's mode is "editor", handle lowercase given the current tab's node's source; exit.

If the current tab's mode is "writer", handle lowercase given the current tab's node's page; exit.

Cluck.

To handle measure:

If the current tab's mode is "writer", handle measure given the current tab's node's page; exit.

Cluck.

To handle minimize:

Minimize the main window.

To handle mirror:

If the current tab's mode is "writer", handle mirror given the current tab's node's page; exit.

Cluck.

To handle new directory:

If the current tab's mode is "finder", handle new directory given the current tab's node; exit.

Cluck.

To handle new document:

If the current tab's mode is "finder", handle new document given the current tab's node; exit.

Cluck.

To handle new page:

If the current tab's mode is "docster", handle new page given the current tab's node's document; exit.

Cluck.

To handle new shape given a kind:

If the current tab's mode is "writer", handle new shape given the current tab's node's page and the kind; exit.

Cluck.

To handle new text file:

If the current tab's mode is "finder", handle new text file given the current tab's node; exit.

Cluck.

To handle open:

If the current tab's mode is "finder", handle open (finder); exit.

If the current tab's mode is "docster", handle open (docster); exit.

Cluck.

To handle open (docster):

Handle open without drawing (docster).

Show the desktop with status.

Flush the event queue.

Refresh the cursor.

To handle open (finder):

If the current tab's mode is not "finder", cluck; exit.

Get some selected path links given the current tab's node.

If the selected path links are empty, show error "I'm not sure what you want me to open."; exit.

Show status "Opening...".

Put the current tab into a tab.

Loop.

Get a path link from the selected path links.

If the path link is nil, break.

Collapse the tab's node to root-level.

Open the tab's node given the path link's string.

Switch the tab to the tab's node.

Switch to the tab.

If the i/o error is not blank, show error the i/o error; break.

If the tab's node should be read-only, set the tab's node to read-only mode.

Find a free tab starting with the tab's next.

If the free tab is nil, break.

Put the free tab into the tab.

Repeat.

Destroy the selected path links.

Show the desktop with status.

Flush the event queue.

Refresh the cursor.

To handle open as dump:

If the current tab's mode is not "finder", cluck; exit.

Put the current tab's node's first selected nodette into a nodette.

If the nodette is nil, show error "I'm not sure what you want me to open."; exit.

If the nodette's kind is not "file", cluck; exit.

Show status "Opening...".

Open the nodette (file as dump).

If the i/o error is not blank, show error the i/o error; exit.

Switch the current tab to the nodette.

Show the desktop with status.

Flush the event queue.

To handle open as text:

If the current tab's mode is not "finder", cluck; exit.

Put the current tab's node's first selected nodette into a nodette.

If the nodette is nil, show error "I'm not sure what you want me to open."; exit.

If the nodette's kind is not "file", cluck; exit.

Show status "Opening...".

Open the nodette (file as text).

If the i/o error is not blank, show error the i/o error; exit.

Switch the current tab to the nodette.

Show the desktop with status.

Flush the event queue.

To handle open without drawing (docster):

Put the current tab's node's document's first selected page into an page.

If the page is nil, show error "I'm not sure what you want me to open."; exit.

Focus on the page in the current tab's node's document.

Put the page into the current tab's node's page.

Open the current tab's node's page.

Switch the current tab to the current tab's node.

To handle outdent:

If the current tab's mode is "editor", handle outdent given the current tab's node's source; exit.

Cluck.

To handle paste:

If the current tab's mode is "finder", handle paste given the current tab's node; exit.

If the current tab's mode is "editor", handle paste given the current tab's node's source; exit.

If the current tab's mode is "docster", handle paste given the current tab's node's document; exit.

If the current tab's mode is "writer", handle paste given the current tab's node's page; exit.

Cluck.

To handle pen given a color:

If the current tab's mode is "writer", handle pen given the current tab's node's page and the color; exit.

Cluck.

To handle print:

If the current tab's mode is "editor", handle print given the current tab's node's source; exit.

If the current tab's mode is "docster", handle print given the current tab's node's document; exit.

If the current tab's mode is "writer", handle print given the current tab's node's page; exit.

Cluck.

To handle quit:

If the user is not sure, exit.

Loop.

Show status "Quiting...".

Get a tab from the tabs.

If the tab is nil, relinquish control; exit.

If the tab's node is not modified, repeat.

Switch to the tab.

Show the desktop.

If the user is not willing to close the tab's node, exit.

Repeat.

To handle radius:

If the current tab's mode is "writer", handle radius given the current tab's node's page; exit.

Cluck.

To handle redo:

If the current tab's mode is "editor", handle redo given the current tab's node's source; exit.

If the current tab's mode is "writer", handle redo given the current tab's node's page; exit.

Cluck.

To handle reduce:

If the current tab's mode is "writer", handle reduce given the current tab's node's page; exit.

Cluck.

To handle rename:

If the current tab's mode is "finder", handle rename given the current tab's node; exit.

Cluck.

To handle renumber pages:

If the current tab's mode is "docster", handle renumber pages given the current tab's node's document; exit.

Cluck.

To handle reverse:

If the current tab's mode is "finder", handle reverse given the current tab's node; exit.

If the current tab's mode is "editor", handle reverse given the current tab's node's source; exit.

Cluck.

To handle rotate page:

If the current tab's mode is "writer", handle rotate page given the current tab's node's page; exit.

Cluck.

To handle rotate shape:

If the current tab's mode is "writer", handle rotate shape given the current tab's node's page; exit.

Cluck.

To handle run:

If the current tab's mode is "finder", handle run given the current tab's node; exit.

If the current tab's mode is not "editor", cluck; exit.

Handle save all.

If the i/o error is not blank, exit.

Find a directory to compile.

If the directory is blank, show error "I'm not sure what you want me to compile."; exit.

Compile the directory.

If the compiler's abort flag is set, handle compile or list or run error; exit.

Show status "Running...".

Start the current process given the compiler's exe path.

If the i/o error is not blank, show error the i/o error; exit.

Show status the compiler's timer's string then " ms".

To handle save:

If the current tab's node is not savable, cluck; show status "Unable to save"; exit.

Show status "Saving...".

Save the current tab's node.

If the i/o error is not blank, show error the i/o error; exit.

Show status "Saved".

To handle save all:

Show status "Saving...".

Put 0 into a count.

Clear the i/o error.

Loop.

Get a tab from the tabs.

If the tab is nil, show status the count and "file saved" or "files saved"; exit.

If the tab's node is not savable, repeat.

Save the tab's node.

If the i/o error is blank, add 1 to the count; repeat.

Switch to the tab.

Show the desktop.

Show error the i/o error.

To handle save as pdf:

If the current tab's mode is "docster", handle save as pdf given the current tab's node's document; exit.

Cluck.

To handle select all:

If the current tab's mode is "finder", handle select all given the current tab's node; exit.

If the current tab's mode is "editor", handle select all given the current tab's node's source; exit.

If the current tab's mode is "docster", handle select all given the current tab's node's document; exit.

If the current tab's mode is "writer", handle select all given the current tab's node's page; exit.

Cluck.

To handle send to back:

If the current tab's mode is "writer", handle send to back given the current tab's node's page; exit.

Cluck.

To handle smooth:

If the current tab's mode is "writer", handle smooth given the current tab's node's page; exit.

Cluck.

To handle sort definitions:

If the current tab's mode is "editor", handle sort definitions given the current tab's node's source; exit.

Cluck.

To handle sort nodettes given a string:

If the current tab's mode is "finder", handle sort nodettes given the current tab's node and the string; exit.

Cluck.

To handle sort selected lines:

If the current tab's mode is "editor", handle sort selected lines given the current tab's node's source; exit.

Cluck.

To handle stop:

If the current process is 0, cluck; exit.

Show status "Stopping...".

Stop the current process.

Show status "Stopped".

To handle uncomment:

If the current tab's mode is "editor", handle uncomment given the current tab's node's source; exit.

Cluck.

To handle undo:

If the current tab's mode is "editor", handle undo given the current tab's node's source; exit.

If the current tab's mode is "writer", handle undo given the current tab's node's page; exit.

Cluck.

To handle ungroup:

If the current tab's mode is "writer", handle ungroup given the current tab's node's page; exit.

Cluck.

To handle unlabel:

If the current tab's mode is "writer", handle unlabel given the current tab's node's page; exit.

Cluck.

To handle uppercase:

If the current tab's mode is "editor", handle uppercase given the current tab's node's source; exit.

If the current tab's mode is "writer", handle uppercase given the current tab's node's page; exit.

Cluck.

To handle version:

Show status the module's name.

To handle yank given a number:

If the current tab's mode is "writer", handle yank given the current tab's node's page and the number; exit.

Cluck.

A handler is a string.

To initialize:

Initialize the desktop.

Initialize the finder.

Initialize the editor.

Initialize the writer.

Initialize the compiler.

Fill the tabs.

Switch to the tabs' first.

To initialize the bar:

Put "normal" into the bar's mode.

Put 1/4 inch plus 1/3 inch plus 1/4 inch into the bar's height.

Put the screen's left into the bar's left.

Put the screen's top into the bar's top.

Put the screen's right into the bar's right.

Put the screen's top plus the bar's height into the bar's bottom.

Put the bar's box into the bar's display.

Adjust the bar's display given 1/4 inch and 1/4 inch and -1/4 inch and -1/4 inch.

Destroy the menu.

Create the menu.

Initialize the status.

To initialize the desktop:

Put the lightest gray color into the desktop color.

Put the light sky color into the comment color.

Put the lightest sky color into the grid color.

Put the red color into the tracer color. \ tracer

Initialize the bar.

Initialize the switch.

Initialize the pad. \ uses the bar and the switch

To initialize the pad:

Put the screen's left into the pad's left.

Put the bar's bottom into the pad's top.

Put the screen's right into the pad's right.

Put the switch's top into the pad's bottom.

To initialize the status:

Put the bar's display and 1/12 inch into the status' roundy box.

Put the menu's commands' last's right into the status' left.

To initialize the switch:

Initialize the tabs.

Put 1/4 inch plus 1/3 inch into the switch's height.

Put the screen's left into the switch's left.

Put the screen's bottom minus the switch's height into the switch's top.

Put the screen's right into the switch's right.

Put the screen's bottom into the switch's bottom.

Adjust the tabs.

To initialize some tabs:

Add 1 to a count.

If the count is greater than 10, break.

Create a tab.

Append the tab to the tabs.

Repeat.

The menu is a menu.

A menu is a thing with some commands.

A mode is a string.

The pad has a box.

The palette width is a width equal to 2-1/3 inches.

A prompt is a string.

A question is a string.

To run:

Start up.

Initialize.

Handle events.

Finalize.

Shut down.

The selected command is a command.

A shortcut is a string.

To show the desktop:

Draw the desktop.

Refresh the screen.

To show the desktop with status:

Clear the status' string.

If the current tab's mode is "finder", get the status for the current tab's node.

If the current tab's mode is "editor", get the status for the current tab's node's source.

If the current tab's mode is "docster", get the status for the current tab's node's document.

If the current tab's mode is "writer", get the status for the current tab's node's page.

Show the desktop.

To show the desktop with a string:

Put the string into the status' string.

Show the desktop.

To show error a string:

Create the dialog.

Put "error" into the dialog's kind.

Put the red color into the dialog's pen.

Put the string into the dialog's prompt.

Add a button to the dialog given "M'kay".

Cluck.

Handle the dialog.

Destroy the dialog.

To show the status:

Draw the status.

Refresh the screen given the status' roundy box.

To show status a number and a string or another string:

Format the number and the string or the other string into a third string.

Show status the third string.

To show status a string:

Put the string into the status' string.

Show the status.

The status has a roundy box and a string.

A submenu is a menu.

The submenu width is a width equal to 2 inches.

The swatch width is a width equal to 1/3 inch.

To switch the bar to a mode:

Put the mode into the bar's mode.

Clear the status' string.

Show the desktop.

To switch the bar to a mode without drawing:

Put the mode into the bar's mode.

Clear the status' string.

The switch has a height and a box.

To switch to the predecessor of a tab:

If the tab is nil, exit.

Put the tab's previous into another tab.

If the other tab is nil, put the tabs' last into the other tab.

Switch to the other tab.

To switch to the successor of a tab:

If the tab is nil, exit.

Put the tab's next into another tab.

If the other tab is nil, put the tabs' first into the other tab.

Switch to the other tab.

To switch to a tab:

If the tab is nil, exit.

If the tab is the current tab, exit.

Put the tab into the current tab.

To switch a tab to a node:

If the tab is nil, exit.

If the node is nil, exit.

Put the node into the tab's node.

Put the node's designator into the tab's name.

Lowercase the tab's name.

Format the tab's name to fit in the tab's box given the default font.

Put "finder" into the tab's mode.

If the node's source is not nil, put "editor" into the tab's mode.

If the node's document is not nil, put "docster" into the tab's mode.

If the node's page is not nil, put "writer" into the tab's mode.

A tab is a thing with

a box,

a name,

a mode,

a node.

The tabs are some tabs.

To track the desktop given an event:

If the event is nil, exit.

Loop.

If the mouse's left button is up, break.

Put the mouse's spot into a spot.

If the spot's x is greater than the screen's right, move the main window right; repeat.

If the spot's x is less than the screen's left, move the main window left; repeat.

Repeat.

To track the dialog given an event:

If the dialog is nil, exit.

If the dialog's text is nil, exit.

If the event is nil, exit.

Extend any selection in the dialog's text given the event's spot.

If the event's shift flag is not set, deselect the dialog's text.

Clear a selection.

Loop.

If a flag is set, show the desktop with status.

If the mouse's left button is up, exit.

Put the mouse's spot into a spot.

Autoscroll the dialog's text given the spot and the flag.

Extend any selection in the dialog's text given the spot.

If the dialog's text's selection is the selection, repeat.

Set the flag.

Put the dialog's text's selection into the selection.

Repeat.

To track a tab returning a flag:

Clear the flag.

If the tab is nil, exit.

Loop.

If the mouse's left button is up, exit.

Put the mouse's spot into a spot.

Find another tab given the spot.

If the other tab is nil, repeat.

If the tab is the other tab, repeat.

Set the flag.

Remove the tab from the tabs.

If the other tab's left is less than the tab's left, insert the tab into the tabs before the other tab.

If the other tab's left is greater than the tab's left, insert the tab into the tabs after the other tab.

Adjust the tabs.

Show the desktop.

Repeat.