STYLESHEET

By: Tayloe Stansbury

Unsupported

INTRODUCTION

Stylesheets are collections of menus. These collections pop up all at once in a group. This group does not disappear until all menus in it have been dealt with, and the user signals that he is done.

Stylesheets are intended to be used in situations wherein the computer wants an answer to several related questions all at once. One example is font selection. To select a font, the user needs to specify font family (Classic, Modern, etc.), font size (8 point, 10 point, etc.), and font style (bold, italic, etc.). Rather than prompt for each of these parameters in succession, one could use a stylesheet to prompt for it all at once.

When the stylesheet pops up, it will shade (preselect) default selections (if provided) in each of the menus. The user can either decide that the default selections are OK, or change them to suit his taste. (The default selection mechanism can be used to convey the current state of something the user is trying to change with the stylesheet: for example, the current looks of the text with which the user is dissatisfied.)

When the user is finished, he hits the DONE button and the stylesheet disappears, and the final selections are returned. There is also a RESET button. This is useful if the user has mucked up his selections and would like to reinstate the default selections. Finally, there is an ABORT button that if selected returns NIL from STYLESHEET and is intended to provide the user with a convenient way of aborting the selection. Note: This means that NIL can be returned from a call to STYLESHEET.

Menus in a stylesheet can be set up to accept exactly one selection (like a normal menu), less than two selections, or any number of selections. Menus that need not be filled in (i.e., can accept zero selections) have an attached CLEAR button, which can be used to remove selections made in that menu. Menus that can have more than one selection have an attached ALL button, which can be used to select all the items in the menu.

HOW TO MAKE A STYLESHEET

To create a stylesheet, call

(CREATE.STYLE *Prop1 Value1 Prop2 Value2 ... PropN ValueN*) [Function]

CREATE.STYLE accepts an arbitrary number of property-value pairs. Properties currently recognised are

ITEMS [Style Property]

A list of menus. Most menu format parameters contained in menu records are honored by the stylesheet package. WHENSELECTEDFNs are, of course, ignored.

SELECTIONS [Style Property]

A list of menu items, each one corresponding to a menu in ITEMS. The specified selections will be shaded in the appropriate menu, and will be the default selections. If not specified or too short, it will be filled out with NILs (no selection).

NEED.NOT.FILL.IN [Style Property]

A list of T or NIL or MULTI, each one corresponding to a menu in ITEMS. T indicates that the corresponding menu need not be filled in and will be given a CLEAR button. MULTI indicates that the corresponding menu can have any number of selections and will be given both a CLEAR button and an ALL button. If the list is too short, it will be filled out with NILs. If a single T or NIL or MULTI is given instead of a list, it will be replaced by a list of Ts or NILs or MULTIs, respectively.

TITLE [Style Property]

The title that will be given to the stylesheet. If no title is specified, the stylesheet will not have a title bar.

ITEM.TITLES [Style Property]

A list of strings or atoms to serve as titles over the menus. Items without titles specified will not have titles.

ITEM.TITLE.FONT [Style Property]

A fontdescriptor or other font specification which determines the font item titles will be printed in. If NIL, titles will be printed in DEFAULTFONT.

POSITION [Style Property]

The screen position (of type POSITION) of the lower left-hand corner of the stylesheet. If position is not specified, the function STYLESHEET will prompt for the postion (using GETBOXPOSITION). STYLESHEET will modify positions as necessary to ensure that the entire stylesheet will be on the screen.

Stylesheets can be modified by calling

(STYLE.PROP *Stylesheet Prop Newvalue*) [Function]

STYLE.PROP always returns the old value of the specified property of the specified stylesheet. If Newvalue is provided (even if NIL), it replaces the old value. If not provided, the old value remains. (Just like WINDOWPROP.)

To use the stylesheet thus created, call

(STYLESHEET *Stylesheet*) [Function]

This returns a list of selections the user made from the stylesheet. (If a selection is returned as NIL, that indicates that no selection was made.)

One can determine in advance of displaying a stylesheet how big it will be. (This may help in determining a reasonable screen position for the stylesheet.) The relevant functions are

(STYLESHEET.IMAGEWIDTH *Stylesheet*) [Function]

and

(STYLESHEET.IMAGEHEIGHT *Stylesheet*) [Function]

They return the width and height, respectively, of the stylesheet in pixels.

AN EXAMPLE

The package is located in STYLESHEET and STYLESHEET.DCOM. To familiarize yourself with its workings, you might want to load it and try the following example:

```
(SETQ FONT.STYLE
  (CREATE.STYLE
    'TITLE "Please select a font:"
    'ITEM.TITLES '(Family Size Face)
    'ITEM.TITLE.FONT '(Modern 12)
    'ITEMS
      (LIST
        (CREATE MENU ITEMS ← '(Classic Modern Terminal))
        (CREATE MENU ITEMS ← '(8 9 10 11 12 14))
        (CREATE MENU ITEMS ← '(Regular Bold Italic BoldItalic)))
    'SELECTIONS '(Modern 11 Regular)
    'NEED.NOT.FILL.IN 'T]

(STYLESHEET FONT.STYLE]
```