

## **Free Menu Changes:**

This document describes the incompatible changes from the old version to the new version of Free Menu. This document does not describe any of the new features of Free Menu. Some of the terminology used in these notes is introduced in the Free Menu documentation. You should read the Free Menu documentation first.

The function FREEMENU is used to create a Free Menu, replacing and combining the functions FM.MAKEMENU and FM.FORMATMENU.

## **In the Description of the Free Menu:**

There is no longer a WINDOWPROPS list in the Free Menu Description. Instead, the window properties TITLE and BORDER that used to be set in the WINDOWPROPS list can now be passed to the function FREEMENU. Other window properties (like FM.PROMPTWINDOW) can be set directly after FREEMENU returns the window using the system function WINDOWPROP. See the section in the documentation entitled **Free Menu Window Properties**.

Setting the initial state of an item is now done with the item property INITSTATE in the item description, rather than the STATE property.

## **Free Menu Items:**

3STATE items now have states OFF, NIL, and T (instead of a NEUTRAL state). They appear by default in the NIL state.

STATE items are general purpose items which maintain state, and replace the functionality of NCHOOSE items. To get the functionality of NCHOOSE items, specify the property MENUITEMS (a list of items to go in a popup menu), which instructs the STATE item to popup the menu when it is selected. STATE items do not display their current state by default, like NCHOOSE items used to. Instead, if you want the state displayed in the Free Menu, you have to link the STATE item to a DISPLAY item using a Free Menu Item Link named "DISPLAY". The current state of the STATE item will then automatically be displayed in the specified DISPLAY item. The item properties MENUFONT and MENUTITLE also apply to the popup menu.

NWAY items are declared slightly differently. There is now the notion of an NWay Collection, which is a collection of items acting as a single nway item. The Collection is declared by specifying any number of NWay items, each with the same COLLECTION property. NWay Collections have properties themselves, accessible by the macro FM.NWAYPROPS. These

properties can be specified in property list format as the value of the NWayPROPS Item Property of the first NWay item declared for each Collection. NWay Collections by default cannot be deselected (a state in which no item selected). Setting the Collection property DESELECT to any non-nil value changes this behavior. The state of the NWay Collection is maintained in its STATE property.

EDIT items no longer will stop at the edge of the window. Editing is either restricted by the MAXWIDTH property, or else it is not restricted at all. The EDITSTOP property is obsolete. Starting editing with the Right mouse button causes the item to be cleared first.

EDITSTART items now specify their associated edit item (there can only be one, now) by a Free Menu Item Link named "EDIT" from the EDITSTART item to the EDIT item.

TITLE items are replaced by DISPLAY items, which work the same way.

### **Free Menu Interface functions:**

(FREEMENU DESCRIPTION TITLE BACKGROUND BORDER) replaces FM.MAKEMENU and FM.FORMATMENU.

The desired format is not specified as the value of the FORMAT property in the group's PROPS list.

(FM.GETITEM ID GROUP WINDOW) replaces FM.ITEMFROMID. Searches within GROUP for an item whose ID property is *ID*. *ID* is matched against the item ID and then the item LABEL. If *GROUP* is NIL, the entire menu is searched.

(FM.GETSTATE WINDOW) replaces FM.READSTATE. Returns a property list of the selected item in the menu. This list now also includes the NWay Collections and their selected item.

(FM.CHANGELABEL ITEM NEWLABEL WINDOW UPDATEFLG) **new argument order.**

Now works by rebuilding the item label from scratch, taking the original specification of MAXWIDTH and MAXHEIGHT into account. NEWLABEL can be an atom, string, or bitmap. If UPDATEFLG is set, then the Free Menu Group's regions are recalculated, so that boxed groups will be redisplayed properly.

(FM.CHANGESTATE X NEWSTATE WINDOW) **new argument order.** X is either an item or an NWay Collection ID. NEWSTATE is an appropriate state to the type of item. If an NWay collection, NEWSTATE is the actual item to be selected, or NIL to deselect. Toggle items take either T or NIL as NEWSTATE, and 3STATE items take OFF, NIL, or T, and STATE items take any atom, string, or

bitmap as their new state. For EDIT items, *NEWSTATE* is the new label, and FM.CHANGELABEL is called to change the label of the EDIT item.

(FM.RESETSHAPE WINDOW ALWAYSFLG) replaces FM.FIXSHAPE

(FM.HIGHLIGHTITEM ITEM WINDOW) replaces FM.SHADEITEM and FM.SHADEITEMBM.

FM.HIGHLIGHTITEM will programmatically highlight an item, as specified by its HIGHLIGHT property. The highlighting is temporary, and will be undone by a redisplay or scroll. To programmatically shade an item an arbitrary shade, use the new function FM.SHADE.