

MHT-POPUP v1.1

A Lingo Xtra that creates dynamic Popup Menus

Rev 1.0.0

About MHT-Popup

MHT-Popup is a Lingo Xtra which creates popup menus using the Windows API calls. This document will cover the input, processing, and output of the MHT-Popup Lingo Xtra (via method descriptions). Presently, only 16-bit and 32-bit Windows versions of the Xtra are available; therefore this document will concentrate on the Windows Xtra without further mention of the Macintosh platform. Please see the Director sample movie for a complete implementation of MHT-Popup.

Cost, Use, & Registration

Please refer to the Evaluation and Runtime License Agreement for a complete and formal understanding of the terms and guidelines that apply to the use of MHT-Popup.

MHT-Popup may be incorporated in commercially distributed products for a one-time license fee of \$99.

The unregistered version of MHT-Popup included on this archive may be evaluated free of charge. Its functionality is restricted only when used in a Projector.

After purchasing a license for MHT-Popup, the Registration Code you receive will activate MHT- Popup for the Windows version.

If you wish to license MHT-Popup, please follow these instructions fully:

- Print out the Evaluation and Title License Agreement PDF's using Adobe Acrobat.
- Complete the "Licensee" sections on the first and last pages, filling in the appropriate information:
 1. Check which Xtra(s) you wish to license.
 2. "By", "Name", "Title", and "Effective Date"
- Mail the completed License Agreement to Meetinghouse Technologies, along with the appropriate license fee (check or money order -- credit card purchases are available via the web site only).

- Upon receipt of the payment, Meetinghouse Technologies will review and sign the License Agreement and fax it back, along with enabling the Registration Code for MHT-Popup.

How to Contact Meetinghouse Technologies

- Internet
<http://www.meetinghousetech.com>
- e-mail
mhtxtras@meetinghousetech.com
- US Mail
Meetinghouse Technologies
32 Defense Street
Annapolis, MD 21401
- Phone
(410) 573-9273
- Fax
(410) 573-9302

Technical Information

MHT-Popup's functionality is rooted in the Windows API. Windows provides functionality for the creation and manipulation of menus (both popup and permanent). More information on creation and manipulation of Windows menus can be found by searching the online help in either the Windows 16-bit API or Windows 32-bit API. Some details of the popup menu will be described in this document, but for more information refer to the appropriate Windows API online help.

Popup menus work differently from the generic menus that can be accessed from a window's menu bar. A popup menu is not permanently associated with a window, instead a single call must be made to track the popup menu when it should be displayed. Since this popup menu is being created from the Xtra, which does not explicitly have its own window, it must subclass the calling program's window (Director Movie or Projector executable). The window is sub-classed so the Xtra can intercept all the messages from the popup menu. The most important message to intercept is what item (if any) was selected.

The MHT-Popup methods are all Global because only one popup menu can be displayed at a time, and each menu is created so quickly, that there is no need to have multiple menu instances available. In fact, multiple menu's can be dangerous in 16-bit Windows environments because there is significantly less memory for menus.

When an item is selected from a popup menu, a reference string is returned. A reference string is a string which represents the location of the item selected. This reference string can be passed back to MHT-Popup to get a string representation of the name of the item that was selected. Here is an example of a reference string:

Menu String:

“first;second;third[fourth,fifth];sixth”

Reference String for first:

“1”

Reference String for third:

“3”

Reference String for fifth:

“3;2”

The reference string represents the location of the item based on outermost to innermost menu numbering. Therefore, the item “fifth” is located in the 3rd position in the main menu, and it is in the 2nd position in the sub-menu, hence, its reference string “3;2”.

METHODS

MHTPopupNewMenu

Syntax: MHTPopupNewMenu (menuString)

Purpose: Create the popup menu

Lingo Example: MHTPopupNewPopupMenu(“test1;test2;test3[test3.1]”)

Returns: 0 on failure
1 on success

Technical Description:

MHTPopupNewMenu is the first step in the creation of a popup menu. The *menuString* parameter has the following format:

;	menu item separator
[...]	sub-menu

- **menu section separator**

An example string would be:

“firstMenu;secondMenu[firstSubItem;secondSubItem;-;thirdSubItem];thirdMenu”

MHTPopupNewMenu checks the string to see if it has any characters in it. If the string is empty (‘’) or no string exists (NULL pointer), then the function exists will a return value of zero.

MHTPopupDeleteMenu

Syntax: MHTPopupDeleteMenu ()
Purpose: Free the memory held by the popup menu resource
Lingo Example: MHTPopupDeleteMenu ()
Returns: 0 on failure
1 on success

MHTPopupShowMenu

Syntax: MHTPopupShowMenu (posX, posY)
Purpose: To display a popup menu and return a reference to the selected item (converts the character string into a Director string value).
Lingo Example: set retValue = MHTPopupShowMenu()
Returns: empty string if no menu item was selected
reference string constructed from the item selected

Technical Description:

MHTShowPopupMenu make the appropriate Windows API calls to “track” a popup menu. The coordinates passed to this function represent the coordinates relative to the stage window. All control is transferred to the Xtra’s window procedure which is subclassing the main window procedure. Control is not returned to the next statement until the popup menu is removed from the screen (due to a selection or non-selection).

MHTPopupEnableMenuItem

Syntax: MHTPopupEnableMenuItem (string)
Purpose: To enable the menu item defined by the reference "string".
Lingo Example: MHT-PopupDisableMenuItem(“1;2”)
Returns: 0 on failure
1 on success

MHTPopupDisableMenuItem

Syntax: MHTPopupDisableMenuItem (string)
Purpose: To disable the menu item defined by the reference "string".
Lingo Example: MHTPopupDisableMenuItem("1;2")
Returns: 0 on failure
1 on success

MHTPopupReturnItemName

Syntax: MHTPopupReturnItemName (string)
Purpose: Returns a pop-up menu item name
Lingo Example: MHTPopupReturnItemName("1;2")
Returns: empty string on failure
string containing name of menu item on success

MHTPopupRegister

Syntax: MHTPopupRegister (string code)
Purpose: Register this version of the MHT-Popup Xtra.
Lingo Example: MHTPopupRegister("your code here")
Returns: 0 on failure
1 on success