

PopMenu Xtra Documentation

Draft #9 - 3/1/97

Please Note: This documentation is known to be incomplete and incorrect in some places. We are providing it to you on an interim basis as we complete the final version. Check <http://www.gmatter.com> in the future for updates.

Table of Contents

| | |
|---|----|
| <i>Table of Contents</i> | 1 |
| <i>Overview</i> | 3 |
| <i>The Four Types of Xtras</i> | 3 |
| What is a Sprite Xtra? | 3 |
| What is a Lingo Xtra? | 3 |
| What is a Transition Xtra? | 3 |
| What is a Tool Xtra? | 3 |
| <i>How To Install PopMenu Xtra</i> | 4 |
| <i>Shipping PopMenu Xtra with a Runtime Projector</i> | 5 |
| <i>Using PopMenu Xtra in Authoring Mode From The Xtras Menu</i> | 5 |
| <i>How To Create a PopMenu</i> | 6 |
| Using the Insert Menu | 6 |
| Using Lingo | 6 |
| Registering PopMenu Xtra | 6 |
| Using an Unregistered Copy of PopMenu Xtra | 7 |
| Saving PopMenu Xtra Castmembers | 8 |
| <i>The PopMenu Xtra Tool Interface</i> | 8 |
| The PopMenu About Box | 8 |
| The PopMenu Registration Dialog | 8 |
| The PopMenu Online Help | 8 |
| <i>The PopMenu Xtra Sprite Interface</i> | 9 |
| Accessing a PopMenu from the Score Window | 9 |
| Accessing a PopMenu from the Cast Window | 9 |
| Accessing a PopMenu from the Cast Properties Dialog | 10 |
| CastLib and Member Popups | 10 |
| Display Text | 11 |
| Auto-Align Menu Checkbox | 11 |
| Auto-Select Text Checkbox | 11 |
| Auto-Mark Selections Checkbox | 11 |

| | |
|---|-----------|
| Menu Location Popup | 11 |
| Menu Indicator Popup | 11 |
| Look Popup | 12 |
| Mark Type Popup | 12 |
| Set Forecolor Color.../Set Backcolor Color... Buttons | 12 |
| <i>The PopMenu Xtra Menu Definition</i> | 13 |
| Simple Menu Definition | 13 |
| Heirarchical Menu Definition | 13 |
| Meta Characters in Menu Definitions | 13 |
| Menu Spec versus Menu Text | 13 |
| <i>The PopMenu Xtra Lingo Interface</i> | 14 |
| Overview | 14 |
| Standard Castmember Properties | 14 |
| PopMenu Custom Castmember Properties | 15 |
| PopMenu Reference Properties | 15 |
| Menu Structure Properties | 15 |
| Menu Appearance Properties | 16 |
| Commands | 18 |
| Meta Level Commands | 19 |
| Property Level Commands | 21 |
| Functions | 24 |
| <i>Functional Limitations, Compatibility Issue and Other Information</i> | 25 |
| General | 25 |
| Macintosh only | 25 |
| Windows only | 25 |
| <i>Known Bugs</i> | 26 |
| General | 26 |
| Macintosh only | 26 |
| Windows only | 26 |
| <i>Unimplemented Features</i> | 26 |
| <i>Error Messages and Codes</i> | 27 |
| <i>Credits</i> | 28 |
| Red Eye Software | 28 |
| g/matter, inc. | 28 |
| PrintOMatic Xtra | 28 |
| PopMenu Xtra | 28 |
| Sound Xtra | 28 |
| KILLER Transitions | 28 |
| Trans-X Xtra | 28 |
| Director ToolBox Pro | 28 |
| The MediaBook CD for Director | 28 |
| <i>CONTACT INFORMATION</i> | 30 |
| How To Contact Us | 30 |

COPYRIGHT NOTICES _____ **30**

Appendix A _____ **31**

Display Troubleshooting _____ **31**

Appendix B _____ **32**

Director Eccentricities _____ **32**

Overview

PopMenu Xtra allows you to add simple and hierarchical pop-up menus to your Director projects with minimal effort or Lingo programming. An easy to use interface allows you to define the contents and appearance of your custom menus. PopMenu Xtra will save you substantial development time while enabling you to enhance the look and feel of your multimedia project.

Combined with the use of Shockwave for Director, PopMenu Xtra provides a great solution for easily navigating complex websites.

The Four Types of Xtras

Director 5.0 introduced support for a new type of plug-in which is known as an Xtra. Xtras are available in four different types: Sprite, Lingo, Transition and Tool. PopMenu Xtra is a Sprite Xtra which also provides a Tool Xtra interface for registration, help and credits.

What is a Sprite Xtra?

A Sprite Xtra (or an Asset Xtra as it is sometimes referred to) adds additional functionality to Director through the introduction of custom castmember types which subsequently can be used to add new types of sprites to your multimedia production's stage.

What is a Lingo Xtra?

A Lingo Xtra adds functionality to Director through the introduction of new commands and functions which can be accessed through Lingo. Lingo Xtras are very similar to Xobjects in the way that they add functionality to Director.

What is a Transition Xtra?

A Transition Xtra adds new visual effects functionality to Director to enhance the display of new multimedia elements in a project.

What is a Tool Xtra?

A Tool Xtra adds new functionality to Director through the Xtras menu. Tool Xtras can be written using MOA compliant C or C++ code or you can build a Tool Xtra using Director itself and implementing a movie in a window which is accessed from the Xtras menu.

How To Install PopMenu Xtra

When Director launches, it automatically locates Xtras by looking in two special places:

Macintosh

Location #1: The **Xtras** folder which is in the same location as your Director application.

Location #2: The **Macromedia:Xtras** folder contained in your System folder.

Windows

Location #1: The **Xtras** subdirectory which is in the same location as your Director application.

Location #2: The **Windows\Macromed\Xtras** subdirectory on Windows 3.1 -OR- The **Program Files\Common Files\Macromedia\Xtras** subdirectory on Windows 95 or Windows NT

To make the PopMenu Xtra available to your copy of Director, you simply need to copy the PopMenu Xtra folder/subdirectory contained on the enclosed CD onto your hard disk in the appropriate place as described above.

Macintosh

Copy the folder located on the root of the CD named "PopMenu Xtra™"

Windows

Copy the subdirectory located on the root of the CD named "POPMENU"

You can keep PopMenu Xtra in its own folder/subdirectory (Director will search up to 5 levels deep in the Xtras folder/subdirectory looking for Xtras at launch time.)

Note that there are both 16-bit (Windows 3.1) and 32-bit (Windows 95/NT) versions of PopMenu Xtra contained in the POPMENU folder. You only need one of these to be installed in your Xtras folder depending on which version of Director you plan to use. Leaving both in the folder will not cause any problems.

Shipping PopMenu Xtra with a Runtime Projector

PopMenu Xtra can be shipped as part of a Director multimedia project. To do this, you simply need to copy the PopMenu Xtra that you are using in your project into a special folder/subdirectory named **Xtras**. Then you must place this Xtras folder/subdirectory in the same place as your runtime projector application.

IMPORTANT NOTE: YOU MAY NOT DISTRIBUTE the PopMenu Xtra serial number in any human readable form with your project. The PopMenu Xtra serial number is not redistributable under the licensing agreement that accompanies this software and is a copyright violation.

Note that you only need to distribute one of the two Windows versions of PopMenu Xtra which are provided on the CD, either the 16-bit (Windows 3.1) and 32-bit (Windows 95/NT) versions of PopMenu.

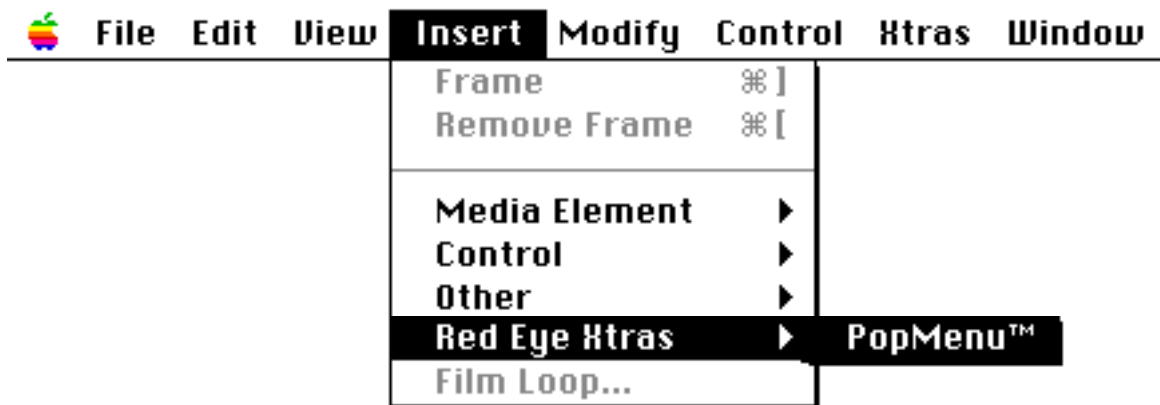
Using PopMenu Xtra in Authoring Mode From The Xtras Menu

PopMenu Xtra can be accessed from a Director movie which is running from the Xtras menu. This means you can build movie in a window (MIAW) authoring time projects that you can distribute to other Director users. You will of course need to have purchased a registered copy of PopMenu Xtra to have the right to re-distribute PopMenu Xtra with your MIAW project.

How To Create a PopMenu

Using the Insert Menu

At any time while authoring in Director, you can quickly and easily create a new PopMenu Xtra castmember by using the Insert menu in Director. Under the Insert menu, you will find the menu choice Red Eye Xtras which will provide a submenu selection of PopMenu Xtra.



Select an empty castmember in one of your cast libraries by opening the Cast window and clicking on an empty cast slot. Now select Insert...Red Eye Xtras...PopMenu Xtra and you will instantly see a new castmember created. You can easily identify a PopMenu castmember by the graphical “thumb” which appears in the Cast window.



Using Lingo

You can create new PopMenu castmembers using Director's Lingo scripting language. Director 5.0 supports a Lingo command named “new” which allows you to insert a new castmember of a specific data type in your cast libraries. Here are a few examples of how to do this:

```
new(#popMenu)
```

```
new(#popMenu, member 100)
```

```
new(#popMenu, member 100 of castLib "menus")
```

Registering PopMenu Xtra

The first time that you create a PopMenu Xtra castmember, you will be prompted to register your copy of PopMenu through the following dialog:

PopMenu Xtra™

Creates Instant Pop - Up Menus in Director

This PopMenu Xtra™ copy is unregistered. Please enter your registration information below. If you choose to Register Later, you will be unable to save your PopMenus.

PopMenu Xtra™
 Entire Program Code
 © 1995-96 Red Eye Software
 Documentation, sample movies
 © 1996 g/matter, inc.

RED EYE SOFTWARE g/matter

Name:

Organization:

Serial number:

If you enter a valid serial number and click on the Register button, then you will have successfully registered your copy of Director with PopMenu Xtra and activated all of PopMenu Xtra's features.

You can also choose to use the Register.. command under the PopMenu Xtra™ menu choice in the Xtras menu to register at any time.

Using an Unregistered Copy of PopMenu Xtra

If you click on the Register Later button, you will be reminded to register later the first time you create a new PopMenu Xtra castmember in each Director session until you actually register. Also, failing to register PopMenu Xtra will cause certain features to be disabled. You won't be able to save properties which you have assigned to PopMenu Xtra castmembers and you won't be able to load previously customized PopMenu Xtra castmembers from disk. All PopMenu Xtra castmembers will revert to default values when you attempt to open a file containing previously customized castmembers.

In addition, using an unregistered version of PopMenu Xtra will result in the following features being disabled:

- setMenu
- setMenuField
- appendItem
- insertItem
- set the menu of
- set the menuField of

Saving PopMenu Xtra Castmembers

It's important to note that when you save your Director file to disk, only the properties associated with your PopMenu Xtra castmembers are saved to disk. Specifically, the status of items in your PopMenu menu definition, such as disablings, marks, etc. are NOT saved to disk. You can capture the state of every menu item through the functions and properties described below and then "manually" reassign them the next time you load the file from disk, if desired.

The PopMenu Xtra Tool Interface

The PopMenu About Box

Selecting the PopMenu-->About... menu command in the Xtras menu will present the following dialog:



The PopMenu Registration Dialog

Selecting the PopMenu-->Register... menu command in the Xtras menu will present the registration dialog as discussed in the section titled "Registering PopMenu Xtra" above.

The PopMenu Online Help

Selecting the PopMenu-->Help menu command in the Xtras menu will open a separate online Help file which contains most of the contents of this documentation in a searchable quick reference format. You must have installed the help file as documented in the section titled "How to Install PopMenu Xtra" above

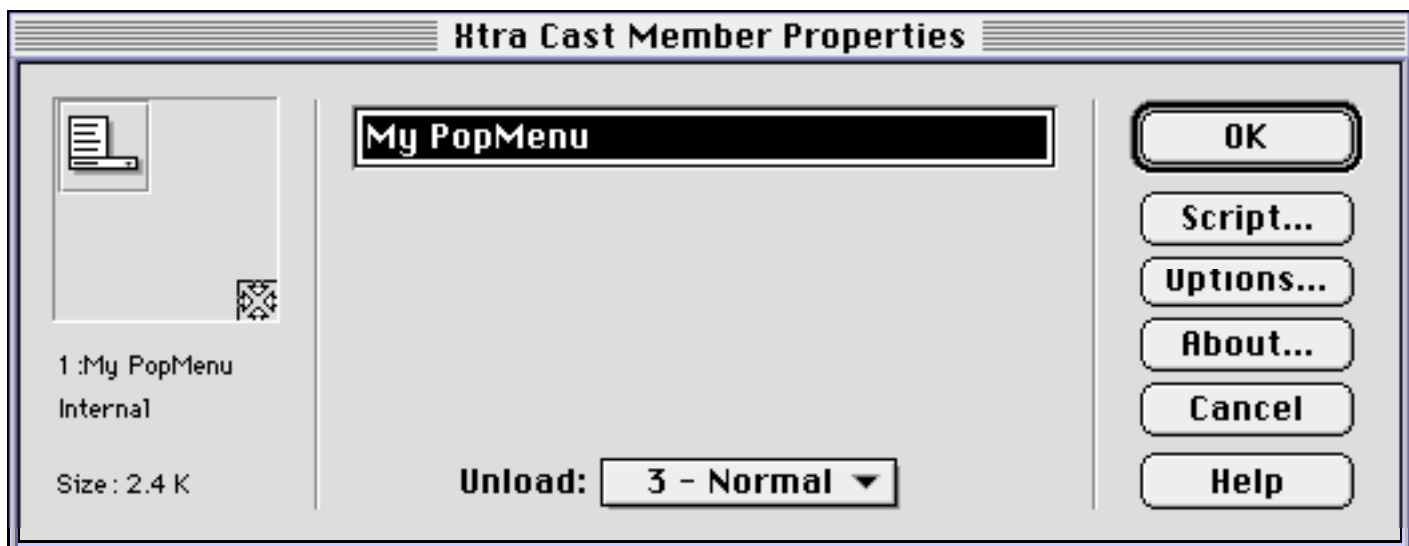
The PopMenu Xtra Sprite Interface

Accessing a PopMenu from the Score Window

Once you have created a PopMenu castmember, you can access it directly from the Score window. Double-clicking the Score cell containing the PopMenu sprite will allow you to access the castmember property dialog for the castmember which that sprite represents. Changes made to the castmember will be reflected by the sprite after closing the castmember properties dialog.

Accessing a PopMenu from the Cast Window

When you click on a PopMenu castmember in Director's Cast window and then select the Info button or the Modify-->Cast Properties menu command, you will see the following properties dialog:



Clicking on the Script... button will allow you to attach a Cast Script to the PopMenu. This is a good way to trap the Mouse Down event and then determine which menu item was last selected by the user.

You can click on the Options... button to display a full PopMenu castmember properties dialog. This dialog allows you to set nearly every property for a particular PopMenu.

The About... button will display the same information dialog that is accessed using the Xtras-->PopMenu->About menu command.

Accessing a PopMenu from the Cast Properties Dialog

This section discusses the functionality of the Cast Properties:Options dialog for a PopMenu castmember.

PopMenu Xtra™

CastLib:

Member:

Menu Field Text:
Not a valid castmember reference.

Menu Field Reference:
Not a valid castmember reference.

Font:

Font Size:

Display Text:

Auto-Align Menu Menu Location:

Auto-Select Text Menu Indicator:

Auto-Mark Selections Look:

Mark Type:

CastLib and Member Popups

The CastLib and Member popups allow you to select the address of a field castmember in your project which contains a valid PopMenu menu definition. You can observe the Lingo equivalent of this selection by looking at the Menu Field Reference area of the dialog where your selection will be referred to as “member x” or “member x of castLib y” in the field.

If a menu reference is invalid or points to an invalid field menu structure, you will see one of the following messages displayed in the Menu Field Text: “Not a valid castmember reference.” or “Missing menu definition (no menu items)”

Font and Font Size Popups and Field

The font name and size popups will allow you to define the appearance of the menu's choices when the menu is displayed as well as the current selection after a menu selection has been made. In addition, you can type in the font size if the size you desired is not displayed in the Font Size popup.

PopMenu Xtra maintains two sets of font and font size settings in each PopMenu Xtra castmember - one for Macintosh and one for Windows. This allows you to create nearly identical display matches between platforms without resorting to using Lingo to constantly change these properties. The display text and the font of the menus themselves will always be of the same face and size. On the Macintosh, the default is 12 point Chicago and on Windows it is 14 point Arial.

Display Text

This editable text line allows you to type in the label text to be displayed to the left of the PopMenu Xtra castmember on the Stage as a sprite.

Auto-Align Menu Checkbox

When this feature is enabled, the menu will center on the text of the last selected menu item automatically the next time it is displayed. If this feature is disabled, the last selected menu item will not be automatically centered - the menu will instead be displayed using the first menu item of the menu.

Auto-Select Text Checkbox

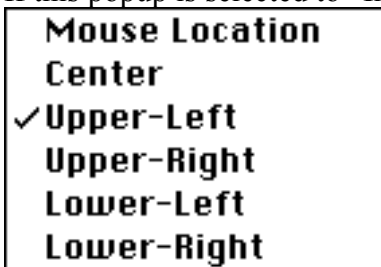
When this feature is enabled, the text of the last selected menu item will be automatically displayed in the display area of the sprite. If this feature is disabled, the display text will not be automatically changed.

Auto-Mark Selections Checkbox

When this feature is enabled, the last selected menu item will be automatically have a mark (based on the Mark Type) displayed next to it the next time the menu is displayed. If this feature is disabled, no menu items will be automatically marked.

Menu Location Popup

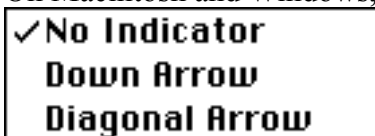
If this popup is selected to "Invisible", then other property options dialog elements will be disabled.



Menu Indicator Popup

This popup allows you to select the type of indicator which will be used with the PopMenu Xtra castmember on the Stage.

On Macintosh and Windows, the supported types are:



Look Popup

If this popup is selected to “Invisible”, then other property options dialog elements will be ignored.



Mark Type Popup

This popup gives you a visual choice of the mark elements which can be used to mark selected menu items.

On the Macintosh and Windows, the supported types are:



The mark type is absolute for all fonts.

Set Forecolor Color.../Set Backcolor Color... Buttons

These buttons allow you to set the associated colors for the PopMenu. Clicking on the button will cause the system color picker to be displayed. You can select an RGB value from the list and the PopMenu will retain that information for later display. NOTE: Please read the Appendix A, “Troubleshooting Display Issues” for information about the impact of palette manipulation when displaying PopMenus.

The PopMenu Xtra Menu Definition

Simple Menu Definition

Menu items are separated through the use of the semi-colon characters “;” or RETURN characters.

Menu example:

item 1;item 2;item 3;item 4

OR

item 1

item 2

...

Using RETURN characters will aid you in creating readable menu definitions in text fields. Note however that if you retrieve the current menu definition (e.g. put the menu of member) it will always return menu items delimited by semi-colons.

Heirarchical Menu Definition

Menu item groups are separated through the user of the left and right square bracket characters. “[“ and “]”.

item 1[subitem 1; subitem 2;subitem 3];item 2[subitem 1; subitem 2;subitem 3];item 3[subitem 1; subitem 2;subitem 3]

Meta Characters in Menu Definitions

| <u>Macintosh</u> | <u>Windows</u> | <u>MetaCharacter</u> | <u>Description</u> |
|------------------|----------------|----------------------|---|
| YES | YES | - | Creates a disabled line separator in the menu at this point. Note that any text found after this metacharacter and before the next item delimited (semi-color or RETURN character) will be ignored. |
| YES | NO | (| Disables the menu item which appears immediately after this character. |
| YES | NO | !√ | Checks the menu item which appears immediately after this character. |
| YES | NO | <B | Bolds the menu item which appears immediately after this character. Ignored on Windows. |
| YES | NO | <I | Italicizes the menu item which appears immediately after this character. Ignored on Windows. |
| YES | NO | <U | Underlines the menu item which appears immediately after this character. Ignored on Windows. |
| YES | NO | <O | Outlines the menu item which appears immediately after this character. Ignored on Windows. |
| YES | NO | / | Places a shortcut key sequence in the menu using the character immediately following the forward slash “/” character. |

Menu Spec versus Menu Text

You can address specific items of a menu using one of two methods. Menu specification allows you to numerically identify the specific location in a menu definition of the item you wish to address. Menu text lets you identify a menu item by its exact text.

Examples:

Menu Specification: 1;3;2 = the second item of the third item of the first item in the menu.

Menu Text: "the menu item" = the first menu item of the exact name or the actual text of the selected item

The PopMenu Xtra Lingo Interface

Overview

PopMenu Xtra supports the full use of Lingo to access PopMenu Xtra castmembers. You can both get and set numerous properties for each castmember. Below is a completion description of each of the specific properties supported by PopMenu Xtra.

NOTE: This will be enhanced with example scripts before ship.

Standard Castmember Properties

`the type of member`

This property returns a value of `#popMenu` for any castmember which has been created using PopMenu Xtra.

```
put the media of member 5
```

```
-- #popMenu
```

`the media of member`

This property returns a copy of the media contained in a PopMenu Xtra. This copy can be assigned as the media of another castmember. The media of member for a PopMenu will contain all of the castmember's custom property data.

```
set the media of member "NewMenu" to the media of member "DefaultMenu"
```

PopupMenu Custom Castmember Properties

PopupMenu Reference Properties

the help of member

This property allows you get a quick help reference for PopMenu Xtra.

the propertyNames of member

This property allows you get immediate access to a listing of all of the properties supported by a PopMenu castmember.

Example:

put the propertyNames of member "MyMenu"

the functionNames of member

This property allows you get immediate access to a listing of all of the functions supported by a PopMenu castmember.

Example:

put the functionNames of member "MyMenu"

Menu Structure Properties

the menu of member

This property allows you to get or set a text string as the menu definition of the PopMenu Xtra castmember.

Example:

set the menu of member "MyMenu" = myFileIO(mReadFile)

set the menu of member "MyMenu" = line 1 to 10 of myMenuDefinition

set the menu of member "MyMenu" = the text of member "MenuDefinition"

put the menu of member "MyMenu"

the menuField of member

This property allows you to get or set a text field (#field) as the menu definition of the PopMenu Xtra castmember.

Example:

set the menuField of member "MyMenu" = member 3

set the menuField of member "MyMenu" = field "MenuDefinition" of castLib 3

the selectedSpec of member

This property allows you to get the item specification of the most recently selected menu item of the PopMenu Xtra castmember. You can ONLY get this property.

The default value is a null string or EMPTY.

the selectedText of member

This property allows you to get the exact text of the most recently selected menu item of the PopMenu Xtra castmember. You can ONLY get this property.

The default value is a null string or EMPTY.

Menu Appearance Properties

the menuAutoMark of member

This property allows you to get or set whether items will be automatically marked upon their selection.

The default value for this property is TRUE.

You can set this property to a TRUE (1) or FALSE (0) value.

the menuAutoText of member

This property allows you to get or set whether selected items' text will be automatically displayed in the PopMenu area upon selection.

The default value for this property is TRUE.

You can set this property to a TRUE (1) or FALSE (0) value.

the menuBackColor of member

This property allows you to get or set the backcolor of the PopMenu Xtra castmember.

The default value for this property is WHITE (0 on Macintosh and 255 on Windows?).

You can set this property to an integer value from 0 to 255 which represents a palette index, not a "true" RGB color. When running in a color depth greater than 8-bits, this value may not be the actual palette index. If you wish to specify RGB values for the properties, use the SetForeColor() and SetBackColor() functions which can passed an RGB value.

the menuDisplayText of member

This property allows you to get or set the text displayed inside the PopMenu Xtra sprite's rectangle on the Stage.

The default value for this property is a null string or EMPTY.

the menuForecolor of member

This property allows you to get or set the forecolor of the PopMenu Xtra castmember.

The default value for this property is BLACK (255 on Macintosh and 0 on Windows?).

You can set this property to an integer value from 0 to 255 which represents a palette index, not a "true" RGB color. When running in a color depth greater than 8-bits, this value may not be the actual palette index. If you wish to specify RGB values for the properties, use the SetForeColor() and SetBackColor() functions which can passed an RGB value.

the menuFont of member

This property allows you to get or set the name of the font used to display the menu of the PopMenu Xtra castmember.

Note: If you specify an invalid font name, PopMenu Xtra will reset the font to the default system font for that computer. Macintosh default font is Chicago 12 point and the Windows default font is specific to the settings of the operating system in authoring mode (usually Helvetica 10 point.)

Note: Setting the menuFont property will always affect the font of the platform you are currently working with.

the menuFontSize of member

This property allows you to get or set the size of the font used to display the menu of the PopMenu Xtra castmember.

Note: Setting the menuFontSize property will always affect the font of the platform you are currently working with.

This will be limited to 96 point on Macintosh and Windows.

the menuIndicator of member

This property allows you to get or set the type of menu indicator which appears on the sprite instance of the PopMenu Xtra castmember.

The default value for this property is 0 or none.

Options are:

- 0 = None
- 1 = Down
- 2 = Diagonal

the menuIndicatorPresent of member

This property allows you to get or set the display of the menu indicator currently assigned to the PopMenu Xtra castmember.

The default value for this property is TRUE.

You can set this property to a TRUE (1) or FALSE (0) value.

the menuLoc of member

This property allows you to get or set an integer value identifying the relative menu location of the PopMenu Xtra castmember.

menuLoc is an integer value from 0 to 5.

Options are:

- 0 = Mouse location
- 1 = Center
- 2 = Upper-left
- 3 = Upper-right
- 4 = Lower-left
- 5 = Lower-right

The default value for this property is 2 or upper-left.

the menuLook of member

This property allows you to get or set an integer value or symbol which identifies the appearance of the clickable PopMenu area of the PopMenu Xtra castmember.

Options are:

0 = Invisible
1 = Macintosh Standard

Commands

In addition to manipulating standard and custom properties of PopMenu Xtra castmembers, you can also use special commands and functions provided by PopMenu Xtra to perform special tasks or to change the values of properties.

What is a Command?

A command is a mechanism which you use in your code to perform a specific task. You may pass a command optional arguments. Generally, you do not expect to get any data returned by a command. If you do, the data will usually represent the success of the command itself in completing its assigned task.

Examples:

```
executeMyCommand
executeMyCommand myData
put executeMyCommand(myData) into successResult
```

What is a Function?

A function is a mechanism which you use in your code to evaluate an expression and return a calculated result. You will usually pass a function arguments although sometimes a function will not require them. You will always expect to get data returned by a function. The data will usually represent the result of evaluating the data passed as arguments to the function itself when it was initially called.

To get a function to perform its work and return a result, you need to surround the argument(s) with a set of parenthesis. Even if the function requires or accepts no arguments, you still need to use the parenthesis to get Director to return the function value. You can also use the special keyword "the result" to get the last function result calculated.

Examples:

```
put myFunction(myData) into calculationResult
put myFunction(myExpression) into calculationResult
put myStaticFunction() into functionResult

myFunction(myData)
put the result into calculationResult
```

Using functions over properties has an advantage in that functions all return error codes with a zero indicating no error.

Meta Level Commands

`setMenu member memberID, string menuText`

This command allows you to set the text of a specific PopMenu Xtra castmember.

Equivalent to set the menuText of member.

`getMenu member memberID`

This command allows you to get the text of a specific PopMenu Xtra castmember.

Equivalent to put the menuText of member.

`setMenuField member memberID, member fieldID`

This command allows you to set the field castmember contains the definition of a specific PopMenu Xtra castmember.

Equivalent to set the menuField of member.

`getMenuField member memberID`

This function returns the field castmember containing the definition of a specific PopMenu Xtra castmember.

`clearMenu member memberID`

This command allows you to clear the entire menu definition for a specific PopMenu Xtra castmember.

Equivalent to:

`setMenu member memberID, ""`

`set the menuText of member memberID = ""`

`refreshMenu member memberID`

This command allows you to refresh the menu definition for a specific PopMenu Xtra castmember based on the current setting for the menuField property. If there is no menuField property value then no change will be made to the menu definition that was active.

Tip: Don't call this command inside of a repeat loop. It takes a little extra time to process.

`appendItem member memberID, string menuspec, string menutext`

This command allows you to append a menu item to the menu definition of a PopMenu Xtra castmember.

`insertItem member memberID, string menuspec, string menutext`

This command allows you to insert a menu item in the menu definition of a PopMenu Xtra castmember.

`deleteItem member memberID, string menuspec`

This command allows you to delete a menu item from the menu definition of a PopMenu Xtra castmember.

`disableItem member memberID, string menuspec`

This command allows you to disable a menu item in the menu definition of a PopMenu Xtra castmember.

Note: On Macintosh systems you cannot disable the 32nd or later menu items. This is a restriction imposed by the Macintosh OS.

`enableItem member memberID, string menuspec`

This command allows you to enable a menu item in the menu definition of a PopMenu Xtra castmember.

`getItem member memberID, string menuspec`

This command allows you to get the menu item stored in a specific place in the menu definition of a PopMenu Xtra castmember.

`setItem member memberID, string menuspec, string menutext`

This command allows you to set the menu item stored in a specific place in the menu definition of a PopMenu Xtra castmember.

`clearMarks member memberID`

This command allows you to clear all marks stored for the selected menu items of a PopMenu Xtra castmember.

Property Level Commands

`setItemMark member memberID, string menuspec, markType`

This command allows you to set the item mark stored for the specified menu item of a PopMenu Xtra castmember.

Options:

<insert mark type integer table here>

`getItemMark member memberID`

TO BE WRITTEN

`setAutoAlign member memberID, boolean`

This command allows you to set a boolean (TRUE or FALSE) which identifies whether or not the menus are to be automatically aligned or not.

`getAutoAlign member memberID, boolean`

This function returns the auto-alignment setting for the menus.

`setAutoMark member memberID, boolean`

This command allows you to set a boolean (TRUE or FALSE) which identifies whether or not marks are to be used in the menu of the PopMenu Xtra castmember.

This is the equivalent to setting the menuAutoMark of member property.

`getAutoMark member memberID`

TO BE WRITTEN

`setAutoText member memberID, boolean`

This command allows you to set a boolean (TRUE or FALSE) which identifies whether or not the last selection item should be automatically displayed within the sprite rectangle of the PopMenu Xtra castmember on the Stage.

This is the equivalent to setting the menuAutoText of member property.

`getAutoText member memberID`

TO BE WRITTEN

`setDisplayText member memberID, string`

This command allows you to set a string which will be displayed within the sprite rectangle of the PopMenu Xtra castmember on the Stage.

This is the equivalent to setting the menuDisplayText of member property.

`getDisplayText member memberID`

This function returns the menuDisplayText of member property.

setIndicatorPresent member memberID, boolean

This command allows you to set a boolean which identifies whether or not a menu indicator will be used in the menu of the PopMenu Xtra castmember.

This is the equivalent to setting the menuIndicatorPresent of member property.

setIndicator member memberID, integer

This command allows you to set an integer which identifies the type of menu indicator to be used in the menu of the PopMenu Xtra castmember.

This is the equivalent to setting the menuIndicator of member property.

getIndicator member memberID

This function returns the menuIndicator of member property.

setFont member memberID, fontname

This command allows you to set a string value which identifies the name of the font to use in the menu of the PopMenu Xtra castmember.

Note: If you specify an invalid font name, PopMenu Xtra will reset the font to the default system font for that computer.

Note: Calling SetFont() will always affect the font of the platform you are currently working with.

This is equivalent to setting the menuFont of member property.

getFont member memberID

This returns the current font name of the specified member.

setFontSize member memberID, size

This command allows you to set an integer value which identifies the size of the text in the menu of the PopMenu Xtra castmember.

This is equivalent to setting the menuFontSize of member property.

This will be limited to 96 point on Macintosh and ?? point on Windows.

getFontSize member memberID

This returns the current font size of the specified member.

setLook member memberID, look

This command allows you to get or set an integer value or symbol which identifies the appearance of the clickable PopMenu area of the PopMenu Xtra castmember.

Options are:

0 = #invisible
1 = #macStandard

This is equivalent to setting the menuLook of member property.

If the look is set to invisible, the PopMenu castmember will NOT display the indicator element for the castmember. In fact, nothing related to the display of the menu will be active.

getLook member memberID

TO BE WRITTEN

setMenuLocation member memberID, integer

This command allows you to set an integer value which identifies the location of the PopMenu relative to the sprite appearance on the Stage.

This is equivalent to setting the menuLoc of member property.

getMenuLocation member memberID

This function returns the menuLoc of member property.

setItemStyle member memberID, string menuspec, styleInteger

(Macintosh only)

This command allows you to specify the style appearance of a specific menu item of the menu displayed by a PopMenu Xtra castmember.

Options:

styleInteger is an integer from 0 to 127 which identifies the specific style to apply to the item. This integer is computed by combining the values identifying individual style components together, e.g. 3 = 1 + 2 (bold + italic)

0 = plain text
1 = bold
2 = italic
4 = underline
8 = outline
16 = shadow
32 = condensed
64 = extended

Examples:

-- set the style of the third item of the first menu item in the menu to plain style

setItemStyle member "MyMenu", "1;3", 0

-- set the style of the "Quit" item of the menu to bold, italic style

setItemStyle member "MyMenu", "Quit", 3

WARNING: This feature may be removed before release.

getItemStyle member memberID, string menuspec

TO BE WRITTEN

setForecolor member memberID, indexValue or rgbValue

This command allows you to specify the foreColor of the menu displayed by a PopMenu Xtra castmember.

Options:

indexValue is an integer from 0 to 255 representing an indexed palette color.

rgbValue is a series of three integer (from 0 to 255) representing the RGB values for a color.

Examples:

setForecolor member "MyMenu", 145

setForecolor member "MyMenu", 16, 255, 123

This is equivalent to setting the menuForecolor of member property.

Note about RGB color versus Indexed color:

RGB colors are generally better to use than Index colors. An RGB color identifies the color in a way which PopMenu can try to maintain regardless of the colordepth of the monitor at a specific time. When PopMenu encounters an RGB color specification for a PopMenu castmember, it will search the list of colors currently available and match the RGB color to the nearest available color in the palette. When PopMenu encounters an Indexed color specification for a PopMenu castmember, it will always use whichever color is identified in that specific index location in the current palette. If the palette or colordepth of the screen is changed between the time you specify the index color value for the PopMenu castmember and the time you display it, you may get a substantially different display result.

setBackcolor member memberID, indexValue or rgbValue

This command allows you to specify the backColor of the menu displayed by a PopMenu Xtra castmember.

Options:

indexValue is an integer from 0 to 255 representing an indexed palette color.

rgbValue is a series of three integer (from 0 to 255) representing the RGB values for a color.

Examples:

`setBackColor` member "MyMenu", 145

`setBackColor` member "MyMenu", 16, 255, 123

This is equivalent to setting the `menuBackColor` of member property.

Note about RGB color versus Indexed color:

RGB colors are generally better to use than Index colors. An RGB color identifies the color in a way which PopMenu can try to maintain regardless of the colordepth of the monitor at a specific time. When PopMenu encounters an RGB color specification for a PopMenu castmember, it will search the list of colors currently available and match the RGB color to the nearest available color in the palette. When PopMenu encounters an Indexed color specification for a PopMenu castmember, it will always use whichever color is identified in that specific index location in the current palette. If the palette or colordepth of the screen is changed between the time you specify the index color value for the PopMenu castmember and the time you display it, you may get a substantially different display result.

Functions

`selectedSpec` member memberID

This function allows you to get the menu item specification for the most recently selected menu item of a PopMenu Xtra castmember.

`selectedText` member memberID

This function allows you to get the menu item text for the most recently selected menu item of a PopMenu Xtra castmember.

`help` member memberID

This function returns the help text for the specified PopMenu Xtra castmember.

`wasLastMenuValid`(member memberID)

This is a boolean (TRUE or FALSE) function which identifies allows you to if the last menu definition assigned to a PopMenu Xtra castmember was in fact valid or not.

Example:

```
set the menuField of member "MyMenu" = member "MyNewMenuDefinition"
```

```
refreshMenu member "MyMenu"
```

```
if NOT wasLastMenuValid(member "MyMenu") then
```

```
    alert "An error was encountered in MyMenu's menu definition."
```

```
end if
```


Functional Limitations, Compatibility Issue and Other Information

General

- You cannot use the AppendItem() or InsertItem() commands with a PopMenu which has submenus. Instead you should use GetMenu() to retrieve the menu text, change it via Lingo scripts, reassign the menu text using SetMenu() and then call the ReDraw() command.
- There will be no font-mapping support (not built into Moa). Internally, each font will be stored separately. Calling SetFont()/GetFont() or manipulating the font property will always affect the font of the platform you are currently working with. In other words, PopMenu Xtra stores two separate sets of properties for menuFont and menuFontSize to address this.
- Auto-rectangle calculations of the castmember when dragged out onto the stage are performed as well as when any of the properties affecting its display are changed. If you change the sprite rect, the behaviour will no longer occur and the sprite will not be automatically resized based on changes to its associated castmember.
- If you delete a PopMenu castmember and then immediately insert a new one in its place, Director will crash. This is a known Director bug. To get around this, do anything else that will change Director's undo buffer (such as typing a few characters in the message window) after the deletion and then you can insert a new castmember.
- Setting the foreground color and background color as a Lingo property can be done through the menuForeColor and menuBackColor properties. However, when setting a property, you can only apply a single value to it. In the case of the foreground and background color, this represents a palette index, not a "true" RGB color. When running in a color depth greater than 8-bits, this value may not be the actual palette index.

To get around this, we suggest using the functions: SetForeColor() and SetBackColor() which can specify an RGB value.

- Using trails inks and animating a popup menu causes redraw problems.
- Calling SetFont() with an invalid font name or setting the property to an invalid font name will **always** reset the font to the system font.
- If you change the palette of a Director movie while a popup menu sprite is on the stage, the colors of the popup menu will be remapped to the new palette indexes, usually not the desired color. When the sprite redraws itself, it will be drawn in the correct color.

To get around this, you can use the command: Redraw(<PopMenu castmember>)

Macintosh only

- The Macintosh has a limit of 5 levels of menus. If there is a hierarchy of more than 5 menus, the 6th, 7th, etc. will still be created, however, activating it will show a menu indicator where these menus should be. If you try access menus in the 6th level and below, nothing will happen.
- Menu items greater than 31 cannot be individually disabled. This is a restriction imposed by the Mac OS.

- Macintosh only: The mark type is not absolute for all fonts, e.g. if using Symbol, there is not bullet type. If using Helvetica, the checkmark translates to nothing.

Windows only

None at this time.

Known Bugs

General

- Cast member references are reset to their original value if you press the "Options" button, to bring up the properties editor then press "Done", and then press "Options" once more.
- Sometimes the menu selection does not get drawn properly although querying the selectedText or selectedSpec property will always indicate the selection made by the user. To get around this, place a Redraw() command in the mouseUp script of the castmember such as:

Redraw(member "places list")

Macintosh only

- None known.

Windows only

- None known.

Unimplemented Features

- None known.

Error Messages and Codes

Error codes (mostly internal)

* = definitely can be passed back to user

+ = possibly passed back to user

| <u>Code</u> | <u>Description/reason</u> |
|-------------|---|
| -8001 | Dialog resource not found |
| -8002 | Out of memory |
| -8003 | Invalid object type (internal) |
| -8008 | Unknown error (internal) |
| -8005 | Property not found |
| -8006 | Nil pointer |
| -8007 | Dialog item not found |
| -8008 | No active dialog |
| -8009 | Handle not locked |
| -8010 | Object not found |
| -8011 | List is empty |
| -8012 | String resource not found |
| -8013 | Invalid look type |
| -8014? | Not yet supported |
| -8015 | Invalid type |
| -8016 | Properties not allocated |
| -8017 | Properties too large for stream |
| -8018 | Buffer too small |
| -8019 | Bad parameter* |
| -8020 | Expired version* |
| -8021 | Internal error |
| -8022 | Value out of range* |
| -8023 | Wrong number of parameters* |
| -8024 | Bad parameter type* |
| -8030 | Unregistered* |
| -8221 | File not found* |
| -8501 | Ran out of menus (Macintosh only)* |
| -8502 | This is a submenu+ |
| -8503 | Parsed string is empty+ |
| -8508 | Unable to parse menu string (bracket mismatch)* |
| -8505 | Invalid mark type* |
| -8506 | Invalid menu item* |
| -8507 | Invalid item style* |
| -10101 | Out of castmember space |

Credits

PopMenu Xtra was created by Red Eye Software of San Francisco, California, USA and is exclusively published by g/matter, inc. of San Francisco, California, USA.

Red Eye Software

Scott Kildall is the author of this outstanding software. Bow to his amazing prowess! Email him examples of your slavish devotion at: redeye@igc.org

or get the most recent information about Red Eye Software at their throne, er, website:

<http://www.halcyon.com/redeye/>

g/matter, inc.

g/matter, inc. is the leading publisher of information and tools for mastering multimedia. Established in 1992, we have been responsible for the development of over 70 CD-ROM titles. Since 1994 we have been creating a large number of add-on products for Macromedia software, including:

PrintOMatic Xtra

PrintOMatic is the premier printing tool for Director. The PrintOMatic Xtra adds a full set of page-layout, text and graphics printing features to Macromedia Director projects on Macintosh and Windows. PrintOMatic includes commands for drawing styled text, graphic primitives, bitmapped pictures, and snapshots of the Director stage, all under Lingo control

PopMenu Xtra

PopMenu Xtra adds tools for enhancing your Macromedia Director user interfaces, including pop-up and multi-level heirarchical menus.

Sound Xtra

Sound Xtra adds powerful new sound recording and playback control to your Macromedia Director productions. With Sound Xtra, you can control the use of sound using features which are not available in Lingo or with other Xobjects.

KILLER Transitions

Killer Transitions contains special effects which can be used in your Macromedia Director 5.x and Authorware 3.5 productions to add new visual transitions to your display. Each of the Killer Transitions effects has many powerful settings which can be used to alter the effect to create literally thousands of transition results.

Trans-X Xtra

Trans-X Xtra lets you create your own transition effects by creating animated 1-bit masks and then applying those as a transition. Trans-X ships with over 100 pre-rendered masks to get you started!

Director ToolBox Pro

Director ToolBox Pro contains a powerful suite of development aids for users of Macromedia Director, including cast, score sprite and Lingo analysis, debugging, formatting and management tools which make it easier and faster to produce complete, solid results.

The MediaBook CD for Director

The MediaBook CD for Director is the definitive collection of information and tools for users of Macromedia Director 3, 4 and 5. Learn everything about the Lingo programming language! The MediaBook CD helps you leverage your investment and enrich your use of Director. Professional and novice multimedia producers will save time and money by using the royalty free programming, numerous XObjects for Macintosh and Windows and reuseable, professionally written Lingo provided in The MediaBook CD.

g/matter, inc. also teaches a series of very advanced multimedia development seminars focused on Director, Media Production and Internet development all over the world. Contact us for information on the next seminar near you!

Go to our website for the latest information on all of our multimedia products: <http://www.gmatter.com>

CONTACT INFORMATION

How To Contact Us

If you have questions about using PopMenu Xtra, please contact us at:

support@gmatter.com

If you have suggestions for future features or other products please send email to g/matter, inc at:

ideas@gmatter.com

or send us a facsimile at: +1.415.243.0396

COPYRIGHT NOTICES

Entire program code © 1995-1996 Red Eye Software. All rights reserved.

Documentation and sample movies © 1995-1996 g/matter, inc. All rights reserved.

PopMenu Xtra is a registered trademark of g/matter.

g/matter, MediaBook and The MediaBook CD are registered trademarks and PopMenu Xtra, PrintOMatic Xtra, Sound Xtra, Director ToolBox Pro are trademarks of g/matter, inc. KILLER Transitions is a registered trademark of SharkByte Tools, Inc. SharkByte is a registered trademark of Terry Ranson. Macromedia, Authorware and Director are registered trademarks and Shockwave is a trademark of Macromedia, Inc. Other product names mentioned within this documentation may be trademarks or registered trademarks of other companies.

Appendix A

Display Troubleshooting

Problem

Using trails inks and animating a popup menu causes redraw problems.

Solution

None.

Problem

Changing the active palette in a frame which displays a PopMenu Xtra sprite causes the colors shown by the PopMenu to change.

Solution

If you change the palette of a Director movie while a popup menu sprite is on the stage, the colors of the popup menu will be remapped to the new palette indexes, usually not the desired color. When the sprite redraws itself, it will be drawn in the correct color. To fix this, you can call a special command supported by PopMenu Xtra, `redraw(member)`, which will update the display of a PopMenu Xtra sprite. This problem is due to a known bug in Director 5.0 and 5.0.1 and may go away in the future.

Appendix B

Director Eccentricities

Problem

If you delete a PopMenu castmember and then immediately insert a new one in its place, Director may crash or start deleting sprite references to the recreated castmember. This is a known Director bug. . This problem is due to a known bug in Director 5.0 and 5.0.1 and may go away in the future.

Solution

To get around this, do anything else that will change Director's undo buffer (such as typing a few characters in the message window) after the deletion and then you can insert a new castmember.

>At 11:28 AM 8/12/96, Chris P Zielke wrote:

```
>> ...
>>   There is an apparent bug in Director which can cause the application
>>   to crash if the user takes the following actions:
>>       create an Xtra cast member,
>>       delete the cast member,
>>       recreate the cast member,
>>       and then place the new cast member onto the stage.
>>
>>   This second cast member is told to destroy itself even though the user
>>   has not asked to delete the sprite. Then it continues to be notified
>>   of normal messages for screen updates and such.
>>
```

Yes, this was reported earlier (see below). In order to implement Undo, Director doesn't actually do the deletion until the next operation, and it mistakenly deletes sprite actors of the `_new_` sprite cast member.

The solution is just to do something in between the delete and replace that changes Director's undo buffer, allowing Director to do the deletion before it can get confused with the new cast member.

Director doesn't immediately get rid of the first sprite asset (cast member) that you create, it keeps it around for undo. When Director does get around to ridding itself of the first sprite asset, it appears that Director is deleting every sprite actor (sprite) associated with its slot in the cast, and is mistakenly deleting the sprite actor associated with the replacement.

Your workaround allows Director to get rid of the first sprite asset before it can mix it up with its replacement (if indeed that's what is happening).