Welcome

Welcome to the DWUser.com MaxiMenu.

The DWUser.com MaxiMenu allows you to easily insert highly-customizable menus into your web pages, all directly from Dreamweaver. This guide will step you through using the product.

Introduction to the MaxiMenu Usage Manual

This guide is designed to help you use the MaxiMenu Menu Creator extension for Dreamweaver. To get started, see the Getting Started guide. When you're ready to make manual changes, learn about the Style XML format in the Style XML Guide section. The Skin Creation Guide shows you how to package your style customizations for later use or sharing.

Getting Started With MaxiMenu

This section covers the following aspects of usage:

- Installing the DWUser.com MaxiMenu
- Inserting a new menu into your HTML (or PHP, ASP, etc) page
- Making style customizations to the menu
- Editing an existing menu

To proceed, go to the Installation Section.

Installation

Prerequisites

Prior to installing the MaxiMenu, you must install the free Adobe AIR Runtime. To download and install the AIR Runtime, click the button below:

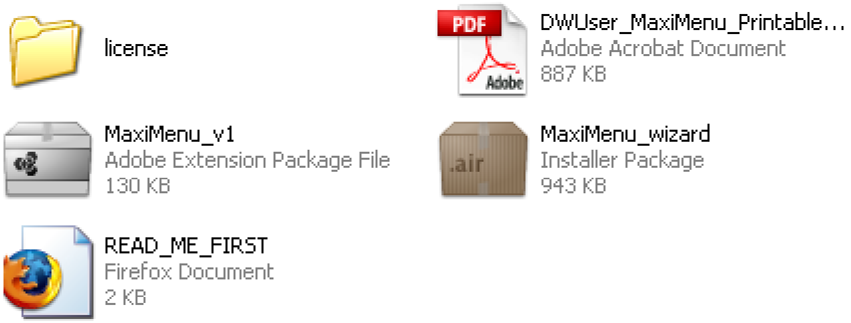
Get ADOBE® AIR™

Once you have installed the AIR Runtime, you can proceed with the installation.

Note: The Adobe (formerly Macromedia) Extension Manager is also required to install the extension file. This is installed automatically with Dreamweaver. In the event you need to upgrade to a more recent version, go to http://www.adobe.com/go/em_download/ .

Download File Overview

When you purchase the software, you will receive a link to the download page. Download the MaxiMenu.zip file from this page. This zip file will contain the following several files:



license

DWUser_MaxiMenu_Printable...
Adobe Acrobat Document
887 KB

MaxiMenu_v1
Adobe Extension Package File
130 KB

MaxiMenu_wizard
Installer Package
943 KB

READ_ME_FIRST
Firefox Document
2 KB

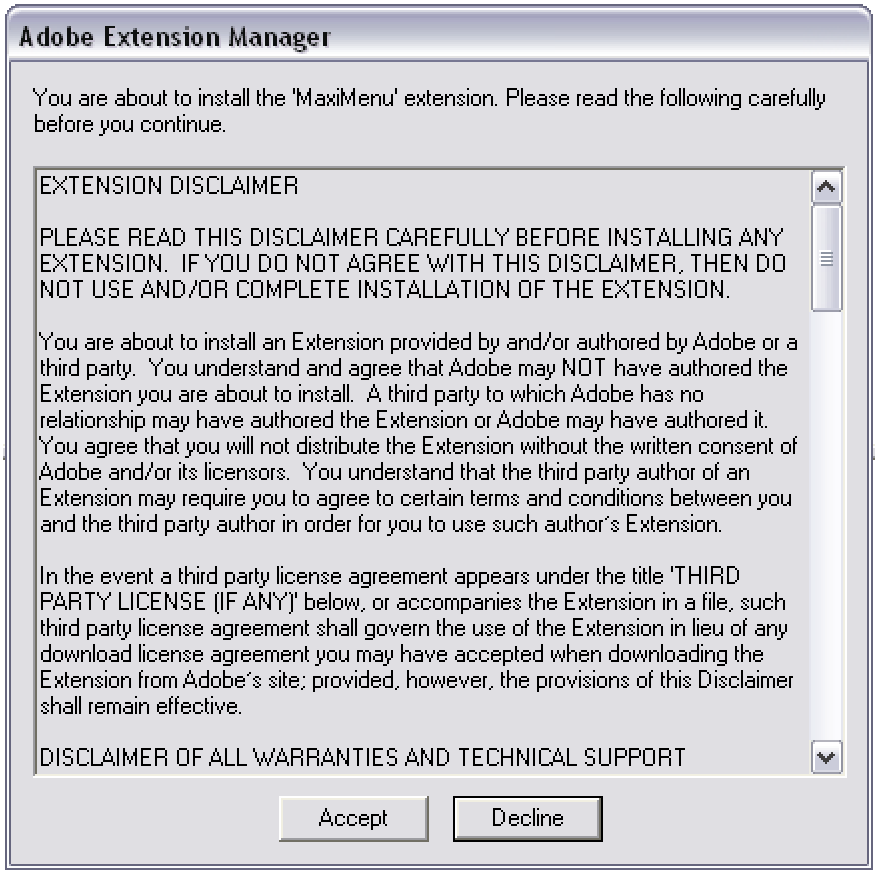
File summary:

- License folder - Contains the user license and build version information
- READ_ME_FIRST.htm - A readme welcome file, with links to web help
- DWUser_MaxiMenu_PrintableHelp.pdf - The printable PDF usage manual
- MaxiMenu_v1.mxp - The Dreamweaver extension file
- MaxiMenu_Wizard.air - The external wizard application installer

Installation

You will need to first install the Dreamweaver extension. To do this, double-click the MaxiMenu_v1.mxp file. This will launch the Extension Manager. You will be prompted to accept the license agreement, then the extension will install.

Windows Vista Users - To ensure proper Dreamweaver extension installation, you may need to run the Extension Manager in Administrator Mode.



You are about to install the 'MaxiMenu' extension. Please read the following carefully before you continue.

EXTENSION DISCLAIMER

PLEASE READ THIS DISCLAIMER CAREFULLY BEFORE INSTALLING ANY EXTENSION. IF YOU DO NOT AGREE WITH THIS DISCLAIMER, THEN DO NOT USE AND/OR COMPLETE INSTALLATION OF THE EXTENSION.

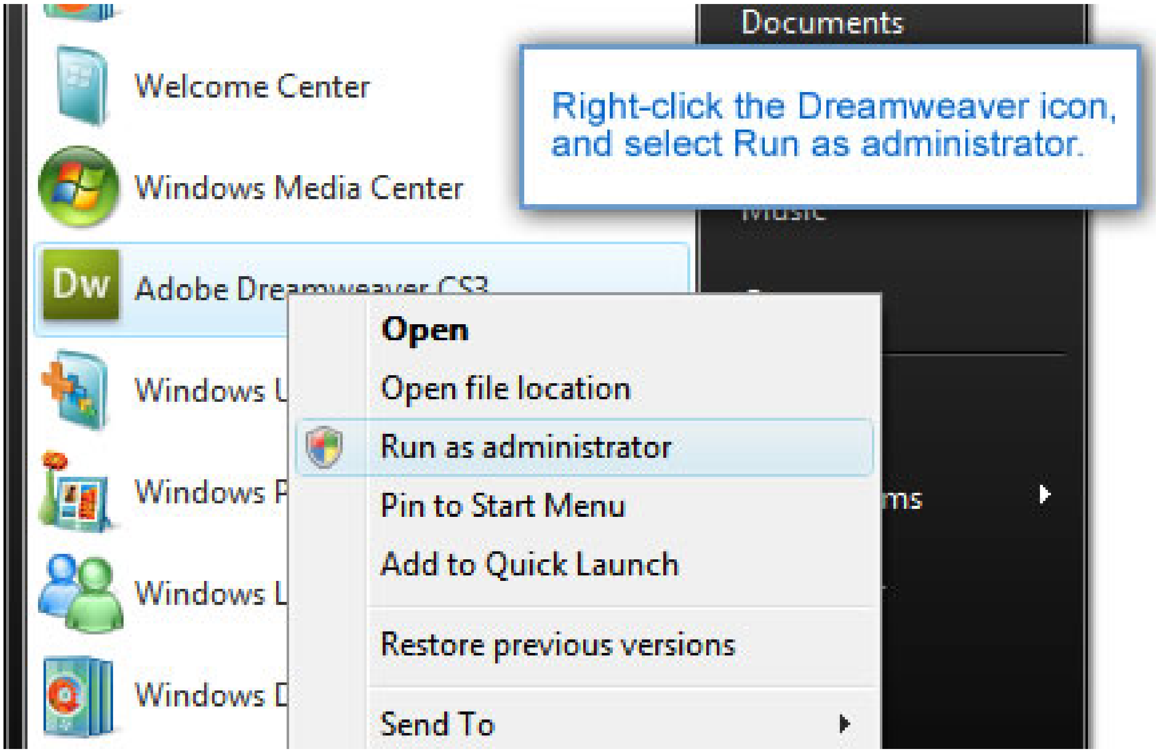
You are about to install an Extension provided by and/or authored by Adobe or a third party. You understand and agree that Adobe may NOT have authored the Extension you are about to install. A third party to which Adobe has no relationship may have authored the Extension or Adobe may have authored it. You agree that you will not distribute the Extension without the written consent of Adobe and/or its licensors. You understand that the third party author of an Extension may require you to agree to certain terms and conditions between you and the third party author in order for you to use such author's Extension.

In the event a third party license agreement appears under the title 'THIRD PARTY LICENSE (IF ANY)' below, or accompanies the Extension in a file, such third party license agreement shall govern the use of the Extension in lieu of any download license agreement you may have accepted when downloading the Extension from Adobe's site; provided, however, the provisions of this Disclaimer shall remain effective.

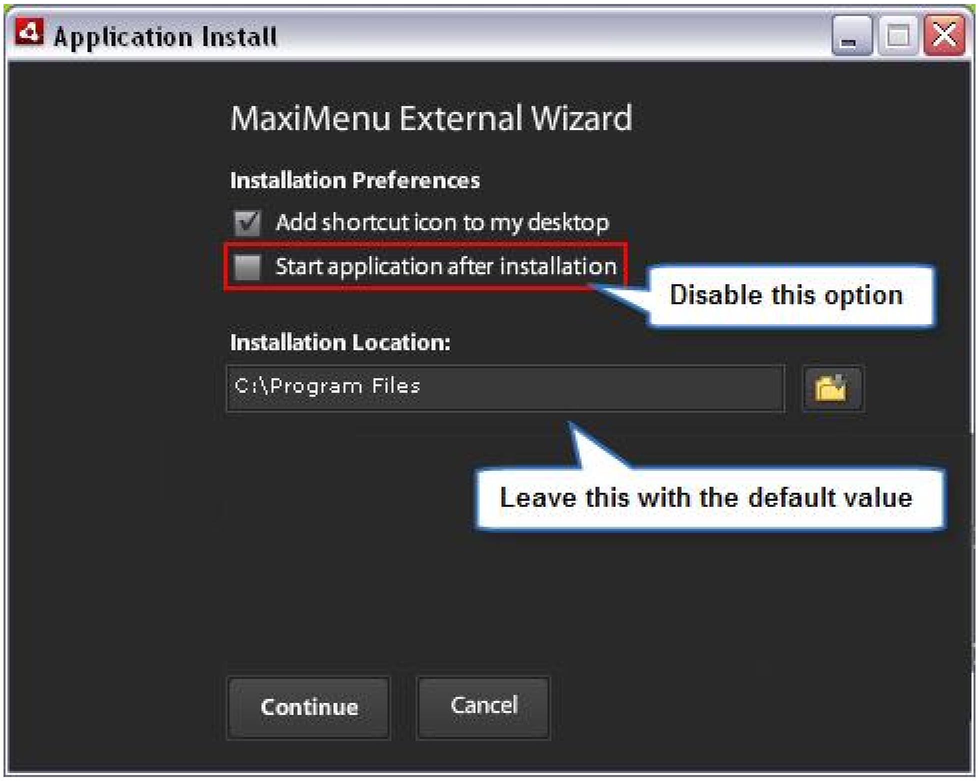
DISCLAIMER OF ALL WARRANTIES AND TECHNICAL SUPPORT

After installation, you must restart Dreamweaver.

IMPORTANT**: If you are using Windows Vista** - you will need to run Dreamweaver as an administrator:


Right-click the Dreamweaver icon, and select Run as administrator.

Next, you need to install the external wizard application. To do this, double-click the MaxiMenu_wizard.air installer file. The AIR installer will guide you through setup. When prompted, disable the 'Launch after Installation' option. The installation location should be your default program folder (this will be automatically filled).


Application Install

MaxiMenu External Wizard

Installation Preferences

☑ Add shortcut icon to my desktop

☐ Start application after installation

Disable this option

Installation Location:

C:\Program Files

Leave this with the default value

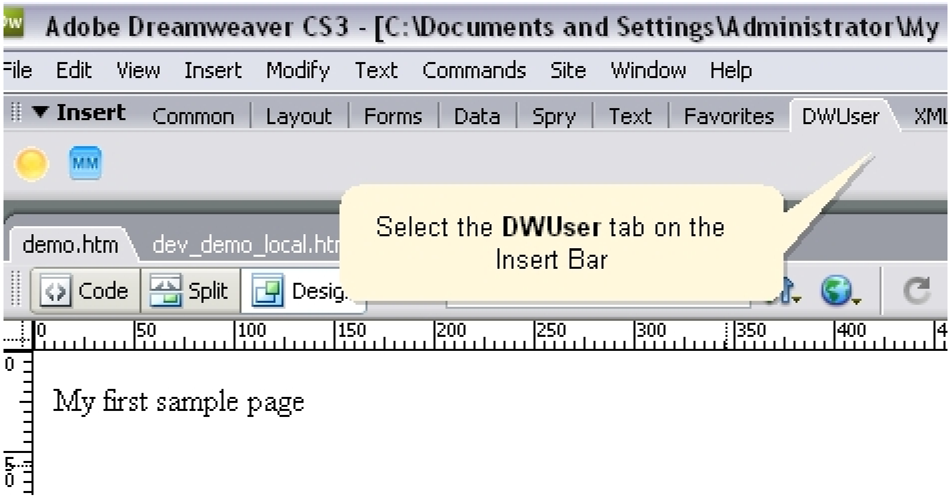
The MaxiMenu is now fully installed, and you are ready to begin using the product. To proceed, go to the Launching the Dreamweaver Extension section.

Launching the Dreamweaver Extension

Launch Dreamweaver. Before adding a menu, you must define a Dreamweaver Site. A Dreamweaver site allows you to specify a specific folder which holds all of your site's files. If you are not familiar with doing this, you can learn about the process by viewing this tutorial on Adobe's website:

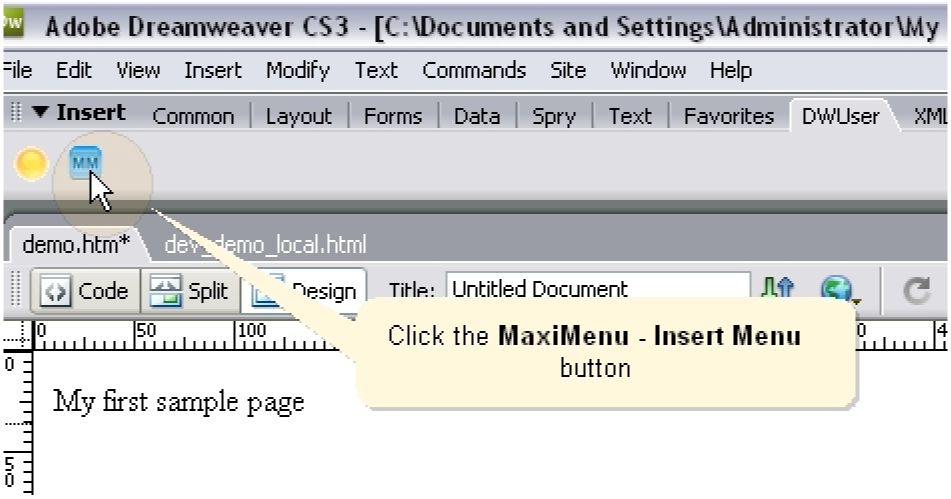
http://www.adobe.com/cfusion/ondemand/index.cfm?id=244&type=2 (note - you many need to allow the popup window to open)

Next, create a new web page (HTML, PHP, ASP, Coldfusion, .NET, etc) or open an existing one. Place your cursor in the location where you want to add the menu. To access the extension, locate the Insert Bar at the top of the Dreamweaver edit window. First, select the DWUser tab:


Adobe Dreamweaver CS3 - [C:\Documents and Settings\Administrator\My
File Edit View Insert Modify Text Commands Site Window Help
Insert Common Layout Forms Data Spry Text Favorites DWUser XML
demo.htm dev_demo_local.htm
Code Split Design
My first sample page

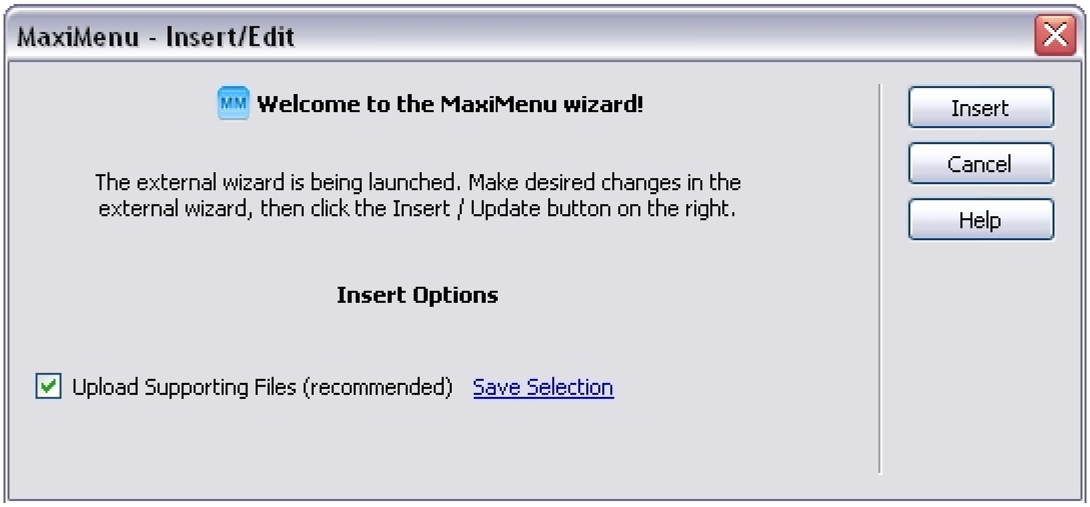
Select the **DWUser** tab on the Insert Bar

Next, click the MaxiMenu icon. This button is labeled 'MaxiMenu - Insert Menu'.


Adobe Dreamweaver CS3 - [C:\Documents and Settings\Administrator\My
File Edit View Insert Modify Text Commands Site Window Help
Insert Common Layout Forms Data Spry Text Favorites DWUser XML
demo.htm* dev_demo_local.html
Code Split Design
Title: Untitled Document
My first sample page

Click the **MaxiMenu - Insert Menu** button

You will momentarily see the Dreamweaver wizard window:



Welcome to the MaxiMenu wizard!

The external wizard is being launched. Make desired changes in the external wizard, then click the Insert / Update button on the right.

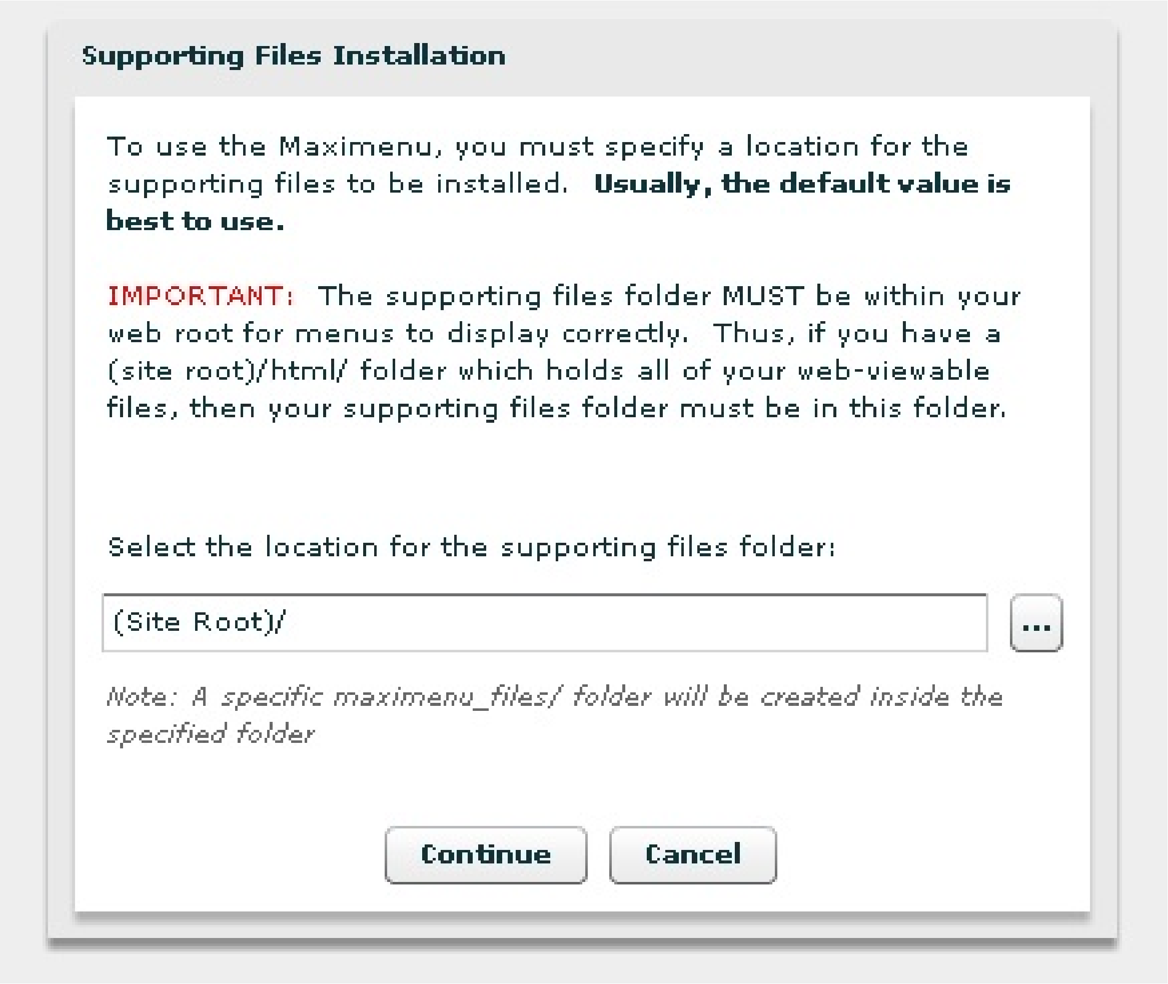
Insert Options

☑ Upload Supporting Files (recommended) Save Selection

This window will only be visible for a second, while the external wizard is initialized. As soon as the external wizard launches, it will become the active window. To proceed, see The Supporting Files Installation Window.

The Supporting Files Installation Window

When the external wizard launches for the first time, you will be prompted to activate your software. Once this process is complete, you will see the Supporting Files Installation Window shown below:



Supporting Files Installation

To use the Maximenu, you must specify a location for the supporting files to be installed. **Usually, the default value is best to use.**

IMPORTANT: The supporting files folder MUST be within your web root for menus to display correctly. Thus, if you have a (site root)/html/ folder which holds all of your web-viewable files, then your supporting files folder must be in this folder.

Select the location for the supporting files folder:

(Site Root)/

Note: A specific maximenu_files/ folder will be created inside the specified folder

To use the MaxiMenu on a site, some supporting files must be copied to the site's folder. This window allows you to specify the location for these files. Usually, the default value of (Site Root)/ is the best option. However, if your web root is inside of the site root (for example, (Site Root)/webroot/ maps to http://www.mydomain.com/), you must specify a location inside of the web root. In this example case, you would specify (Site Root)/webroot/ as the installation location.

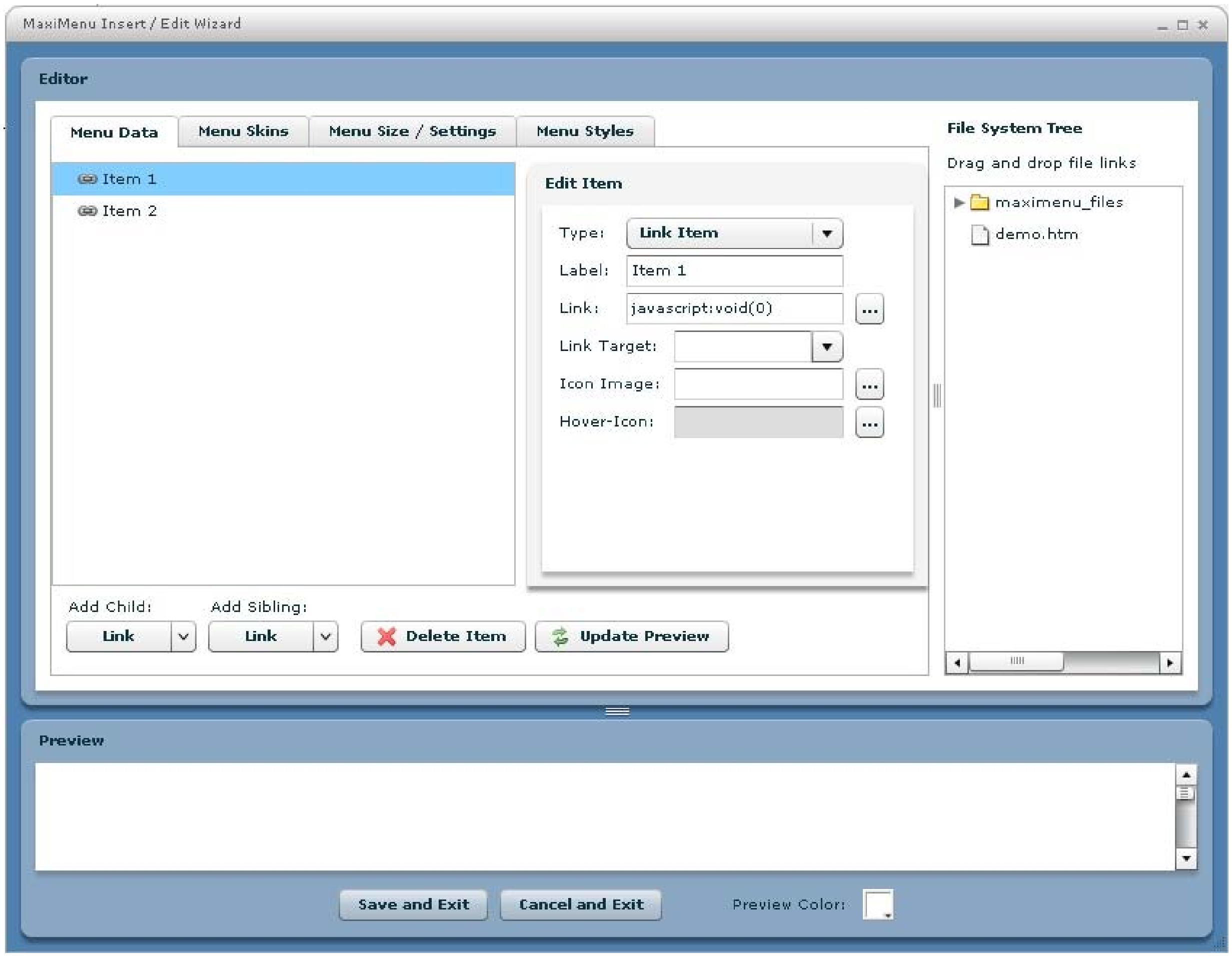
A new maximenu_files/ folder will be created in the location you specify. When you upload the rest of your site, you will need to make sure that this folder is uploaded as well.

Click Continue to install the files and proceed to the main wizard. To learn about the wizard, proceed to the Navigating the External Wizard section.

Note - On future launches of the external wizard, you may be prompted to upgrade the supporting files if a new version is available. If you have made any manual modifications to the supporting files, it is recommended that you back up the existing files prior to allowing the wizard to update the files.

Navigating the External Wizard

The external wizard looks like this:



Type: Link Item

Label: Item 1

Link: javascript:void(0)

Window Control Overview

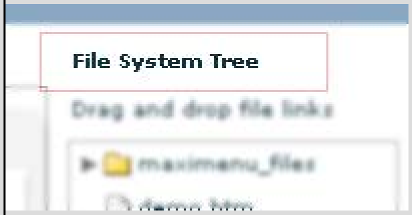
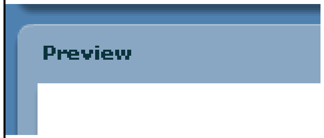
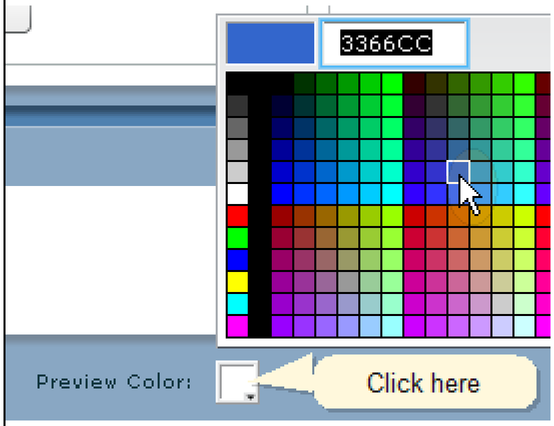
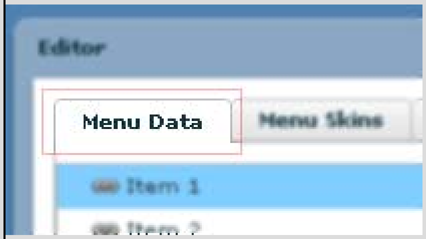
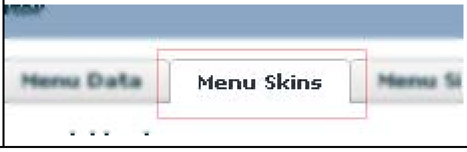
You can use the controls at the top right (top left on Macintosh) to maximize, minimize, or close the window. The title bar can be dragged to move the window. The small resize handle in the far bottom right corner can be used to resize the window. Window size and location data is automatically saved.

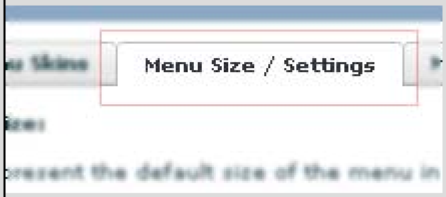
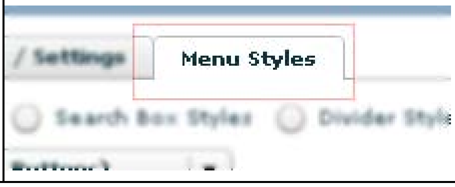
The resize handles between each section (for example, between the tabbed section and the file system tree, or between the Editor panel and the Preview panel) allow you to change the relative section sizes.

The 'Save and Exit' button allows you to save changes and return to Dreamweaver. The 'Cancel and Exit' allows you to discard changes and return to Dreamweaver. If you choose to discard changes, your choice will be confirmed prior to exiting.

Main Section Overview

There are six main sections:

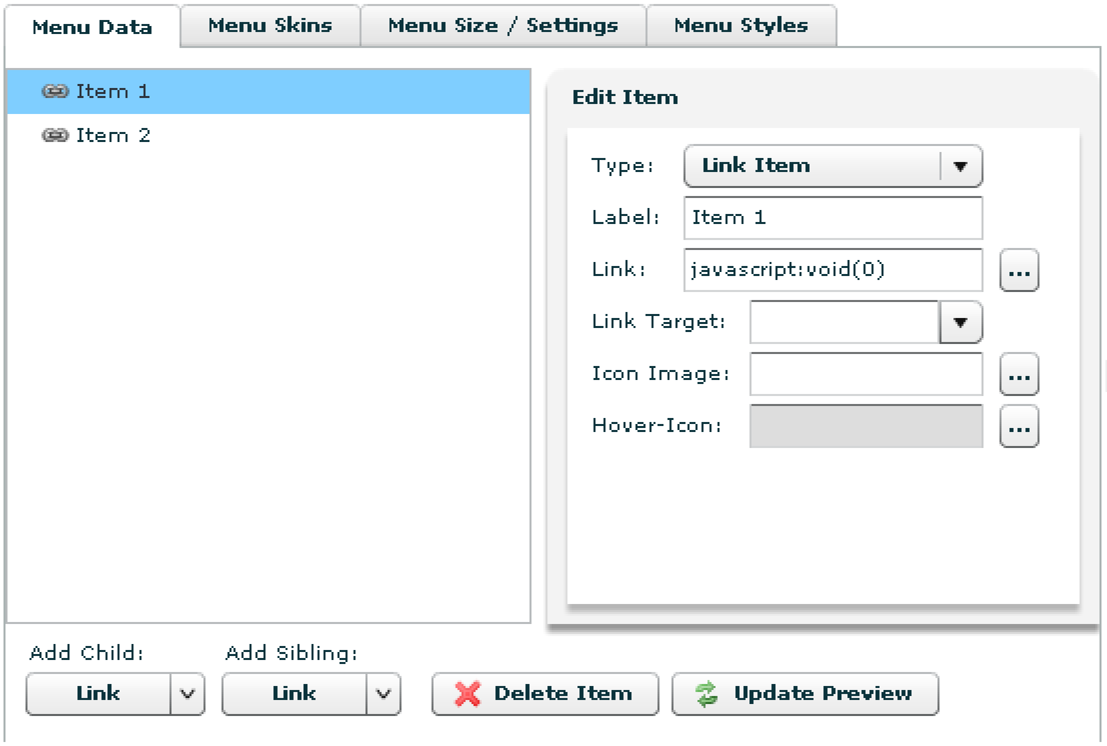
Section Name	Access	Description
File System Tree	Always visible on the right under 'File System Tree':  File System Tree Drag and drop file links maximenu_files demo.htm	This section allows you to explore all files in your Dreamweaver-defined site. Drag and drop a file from the tree into any URL-based field in the wizard. For example, drag a link to a file into the Link Target field for an item on the menu to link to the page.
Preview	Always visible at the bottom of the wizard, in the Preview panel:  	The preview allows you to see how your menu is coming as you make changes. Every time you make a change, you can click the Update Preview button to display a new preview. The Preview Color option at the bottom allows you to specify the background color displayed behind on the preview, allowing you to match your page: 
Menu Data	Select the Menu Data tab at the top of the window:  	This section allows you to specify the data / link structure for your menu. Learn more in Using the Menu Data Section.
Menu Skins	Select the Menu Skins tab at the top of the window:  	This section allows you to select a pre-built style template for your menu. Learn more in Using the Menu Skins Section.

Section Name	**Access**	Description
Menu Size / Settings	Select the Menu Size / Settings tab at the top of the window:  u Skins Menu Size / Settings ize: present the default size of the menu in	This section allows you to modify the size of the menu (standard and expanded), and specify Flash Player detection options. Learn more in [Using the Menu Size / Settings Section]().
Menu Styles	Select the Menu Styles tab at the top of the window:  / Settings Menu Styles Search Box Styles Divider Style	This section allows you to manually edit every single style option for the menu. Learn more in [Using the Menu Styles Section]().

To proceed, go to [Using the Menu Data Section]().

Using the Menu Data Section

The Menu Data section is where you specify the link structure for your menu. The section looks like this:



Type: Link Item
Label: Item 1
Link: javascript:void(0)

The tree displays the current label for each item. If a link has been specified for the item, hovering over the item will display the link in a tooltip. You can **drag items** around inside the tree to change their positions.

Menu Item Types

The following types of menu items can be created:

- **Link Item** - this is a standard link button in the menu.
- **Divider** - this is a button that has no link associated with it, which serves as a denotation of various menu sections (e.g. a divider for Eastern Hemisphere and Western Hemisphere under a Countries menu).
- **Search Box** - this is a search box which allows the user to type in a value and it will be passed as a parameter to the link URL.

Calling Javascript Functions with Item Links

If you want to call a javascript function when the user clicks on an item, you must specify the following format in the link field:

```
javascript:void(functionCalledHere());
```

For example, if you wanted to open a popup window, you would use the following link:

```
javascript:void(window.open('somepage.html', 'newwin',
'toolbars=0,width=400,height=350'));
```

Specifying Search Box Links

When using a Search Box element, you must use a special link to pass the search term in the link opened when the enter key is pressed. To do this, include **{q}**in the link. This value will automatically be replaced with the search/query term.

For example, if you wanted to search Google, use the following link:

```
http://www.google.com/search?q={q}
```

Editing Items

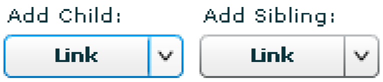
To edit an item, select it by clicking it once. Once the tree item has been selected, you can make modifications in the Edit Item panel to the right of the tree. Changes to the label will be reflected as you type. The fields control the following:

- **Type** - Specifies the type of the item (see above, under Menu Item Types)
- **Label** - Specifies the label displayed on the item
- **Link** - Specifies the link which will be opened when the item is clicked.
- **Link Target** - Specifies the window that the link will be opened in. For example, _blank specifies a new window, and _parent specifies the parent frame.
- **Icon Image** - Specifies the icon image displayed for the item
- **Hover-Icon** - If an Icon Image has been specified, this option allows you to specify the icon displayed when the user hovers over the item.

<u>For Link, Icon Image, and Hover-Icon:</u> You can also drag a link from the File System tree into this field to automatically create a link. Clicking the [...] button to the right of the Link allows you to open a file browse dialog.

Adding and Removing Items

To add item, first select an item which you want to add a new item to (as a child or a sibling). Then use these buttons:



You can only add children to the Link Item button type. A child acts as an item on the parent item's submenu. A sibling acts as another item on the same menu level as other siblings.

By clicking the small arrow on the edge of each drop down menu, you can select which type of item you want to add (see Menu Item Types above).

To remove an item, select the item in the tree. Then, click the Delete Item button:

If the item you are removing has any children, you will be prompted to confirm the deletion.

Update Preview Button

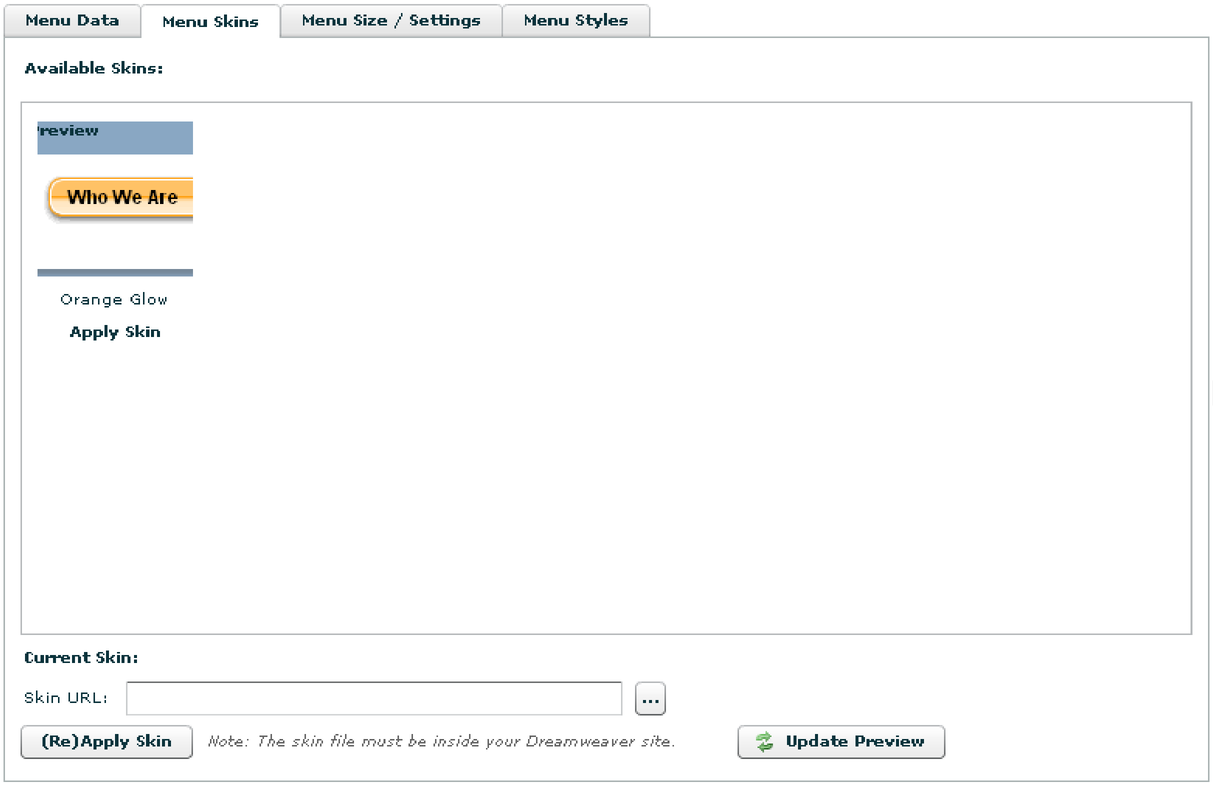
At any time, you can click the Update Preview button to update the preview panel.

Note - If you want to use an external data file (containing static or dynamic content) with your menu, see Using Dynamic Menu Data.

To proceed, go to Using the Menu Skins Section.

Using the Menu Skins Section

The Menu Skins section is where you can specify a pre-built skin template for your menu. The section appears as follows:



Note: The skin file must be inside your Dreamweaver site.

The Available Skins grid displays currently installed skins. The number of skins displayed here will depend on what / how many skin packs you have installed.

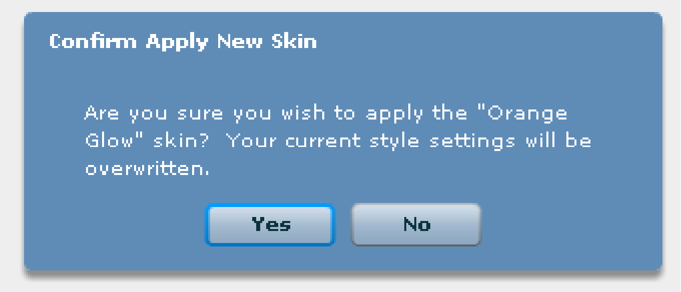
The Current Skin field displays information about the currently applied skin (if any), or allows you to manually select a skin file. If you manually select a skin file, you must then click the (Re)Apply Skin button. Manually specified skins must be located inside the Dreamweaver site.

If you have earlier applied a skin file, then made manual customizations to the styles, you can reset the styles to the skin defaults by clicking the (Re)Apply Skin button.

You can click the Update Preview button at any time to update the preview panel.

Applying a Pre-Built Skin

First, locate the skin you want to use in the Available Skins grid. Next, click the Apply Skin button directly below the skin's name. You will then receive a prompt similar to the following:



Confirm Apply New Skin

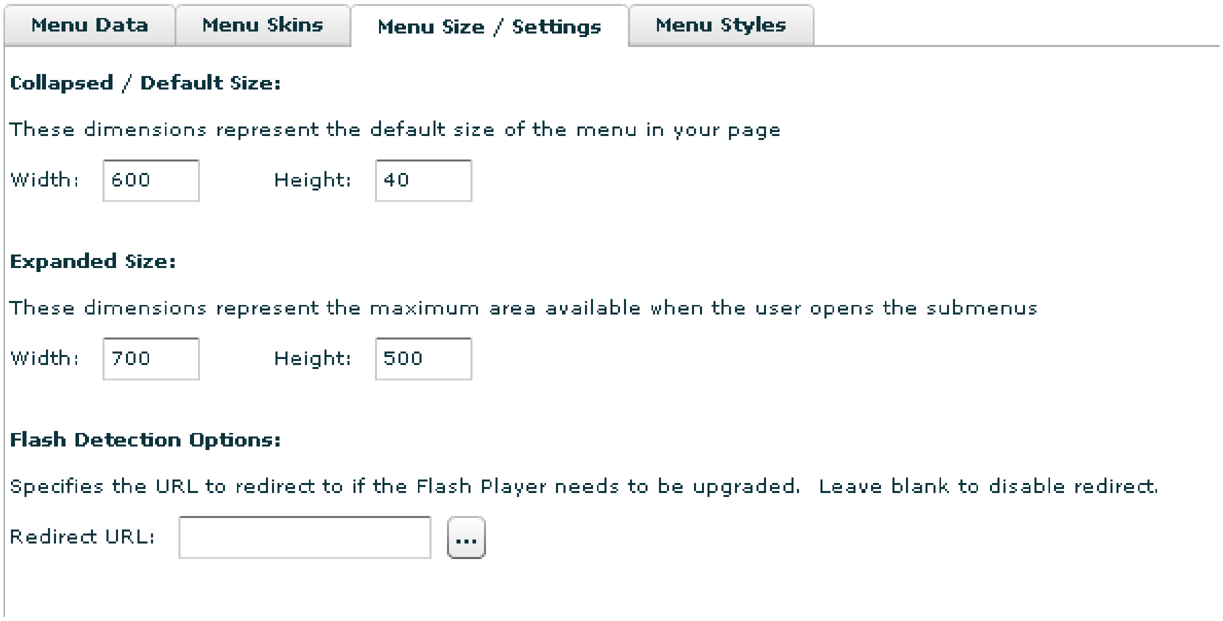
Are you sure you wish to apply the "Orange Glow" skin? Your current style settings will be overwritten.

Click Yes to confirm that you want to apply the skin. The skin will be applied and the styles updated.

To proceed, go to Using the Menu Size and Settings Section.

Using the Menu Size & Settings Section

The Menu Size / Settings section allows you to specify options for the menu dimensions and Flash Player detection. The section looks like this:



Collapsed / Default Size:

These dimensions represent the default size of the menu in your page

Width: 600 Height: 40

Expanded Size:

These dimensions represent the maximum area available when the user opens the submenus

Width: 700 Height: 500

Flash Detection Options:

Specifies the URL to redirect to if the Flash Player needs to be upgraded. Leave blank to disable redirect.

Redirect URL:

Option Overview

The Collapsed/Default Size section allows you to specify the default dimensions of the menu. This is the space the menu occupies in your page.

The Expanded Size section allows you to specify the area available to the menu when the user hovers over the menu to open submenus. Usually the default settings are sufficient.

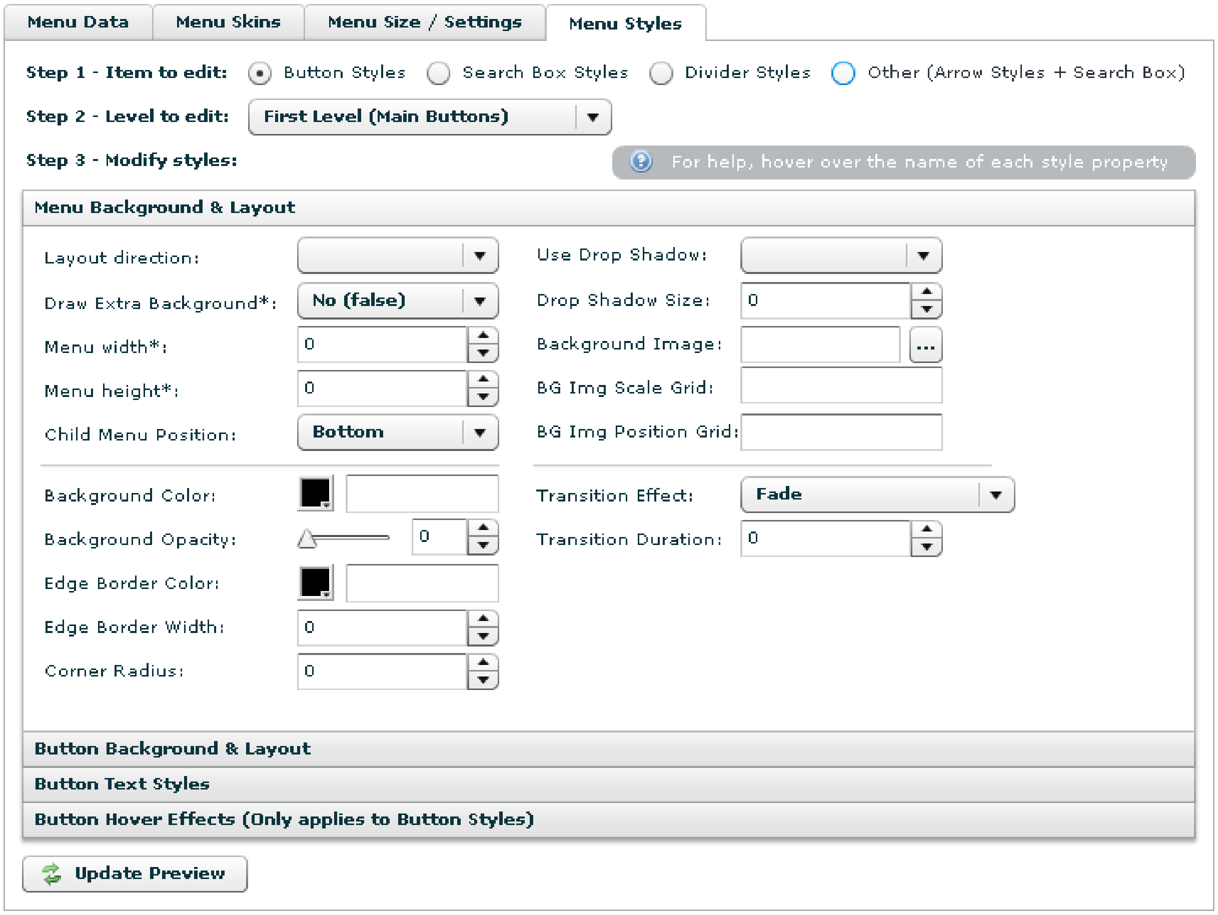
The Redirect URL field under Flash Detection allows you to specify a page to redirect to if the user doesn't have a sufficient Flash Player version installed. Leave the field blank to disable redirect. If you want to disable all Flash Player detection, you can add the following value in your menu's HTML code:

```
MaxiMenu({noDetection:'true',holder_id:...
```

To proceed, go to Using the Menu Styles Section.

Using the Menu Styles Section

The Menu Styles section allows you to customize all of the style options available for the menu. The section appears as follows:


Menu Data
Menu Skins
Menu Size / Settings
Menu Styles
For help, hover over the name of each style property
Menu Background & Layout
Button Background & Layout
Button Text Styles
Button Hover Effects (Only applies to Button Styles)
Update Preview

Step 1 - Item to edit: ● Button Styles ○ Search Box Styles ○ Divider Styles ○ Other (Arrow Styles + Search Box)

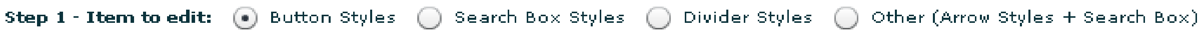
Step 2 - Level to edit: First Level (Main Buttons)

Step 3 - Modify styles:

Layout direction:
Draw Extra Background*: No (false)
Menu width*: 0
Menu height*: 0
Child Menu Position: Bottom
Background Color:
Background Opacity: 0
Edge Border Color:
Edge Border Width: 0
Corner Radius: 0
Use Drop Shadow:
Drop Shadow Size: 0
Background Image:
BG Img Scale Grid:
BG Img Position Grid:
Transition Effect: Fade
Transition Duration: 0

Step 1

To begin, you must select a type of item to edit:



Step 1 - Item to edit: ● Button Styles ○ Search Box Styles ○ Divider Styles ○ Other (Arrow Styles + Search Box)

Select one of the radio buttons to modify that item's styles. The first three options apply to their corresponding menu item types. The 'Other' item controls the submenu indicator arrow and generic search box styles.

Step 2

Next, you must select which menu level you wish to modify. 'First Level' controls the styles for the very first level; these are the main buttons displayed initially. 'Submenu Level 1' controls the styles for the first level of child submenus. Subsequent submenu styles control the submenus for the next levels of child items. You usually only will need to specify values for the first two items, because styles are inherited from one level to the next.

Step 3

Finally, you can modify style values in the accordion-based style panel.

Important tip: To learn about what each option controls, hover your mouse over the label for the style. A tooltip with detailed description information will be displayed. You can also view the detailed Style XML documentation in the Style XML Documentation section of this manual.

Button styles are grouped into four categories:

- **Menu Background & Layout** - Controls general styles for the entire menu level's background and layout. For example, if you are editing the First Level styles, these options control the background displayed behind the main menu.
- **Button Background & Layout** - Controls the button-specific styles for buttons on the current level's menu. For example, controls the border on each individual button.
- **Button Text Styles** - Controls the text styles on the button.
- **Button Hover Effects** - Applies only to Button styles, as Search Boxes and Dividers don't have any hover state. Controls the hover effects displayed, including button background and text styles.

For more detailed documentation of the available style customization options, see the Style XML Documentation section of this manual.

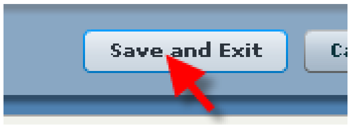
Update Preview Button

At any time as you make edits, you can click the Update Preview button to refresh the preview panel.

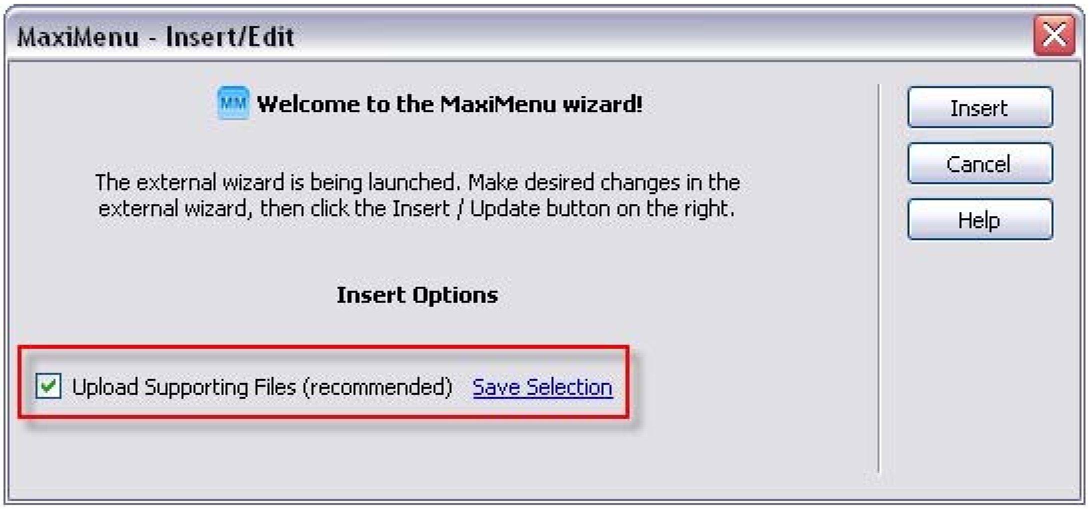
To proceed, go to Inserting the Menu.

Inserting the Menu

Once you have made all desired changes, you are ready to insert the menu. Click the 'Save and Exit' button to return to Dreamweaver:



Back in Dreamweaver, you can enable or disable the auto-upload option:



Welcome to the MaxiMenu wizard!

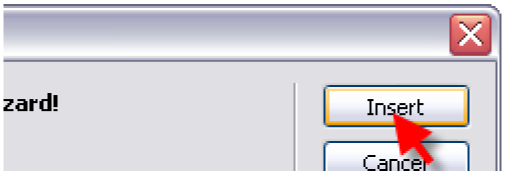
The external wizard is being launched. Make desired changes in the external wizard, then click the Insert / Update button on the right.

Insert Options

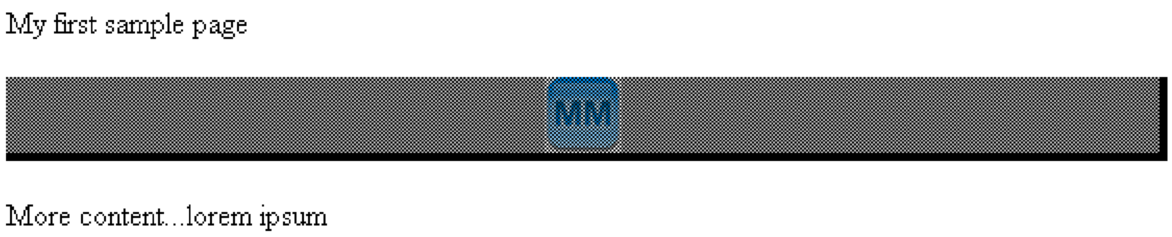
☑ Upload Supporting Files (recommended) Save Selection

Unless you have already defined a remote server for Dreamweaver site, you will want to <u>disable</u> this option. If Dreamweaver is unable to upload the files, you will be notified of this. In this case, you don't need to be concerned; just disable then auto-upload option. Then, when you upload the rest of the files in your site, upload the maximenu_files/ folder as well. Click the Save Selection link to save your preference.

Finally, click the Insert button on the right:



Your new menu will be inserted into your HTML page, and will be visible in the Dreamweaver edit window:



My first sample page

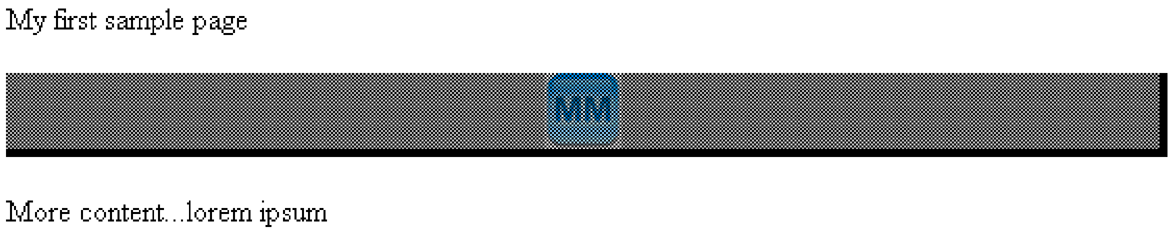
More content...lorem ipsum

To learn about previewing the menu in a browser, see the Previewing in Browser section of the manual.

To proceed, go to the Editing the Menu section.

Editing the Menu

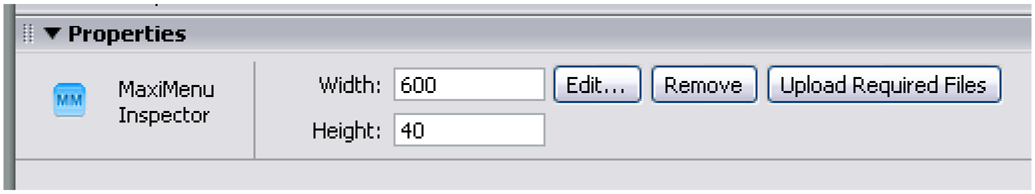
Once you have inserted a menu, select it in Dreamweaver's edit window to make changes:



My first sample page

More content...lorem ipsum

The MaxiMenu Inspector will be visible in the Property Inspector (Window > Properties to open), usually at the bottom of the screen:



Width: 600

Height: 40

You can use this panel to modify the default width and height of the menu (the space it occupies in your page), as well as the following tasks:

Edit - Click this button to launch the edit wizard. The edit wizard functions exactly the same as the insert wizard.

Remove - Click this button to remove the MaxiMenu code from your page. Please note that all supporting files will still remain; these files must be manually removed if you wish to completely erase the menu from the site.

Upload Required Files - This button will attempt to upload the supporting SWF file; however, other files will not be uploaded when using this button. It is best to manually upload the maximenu_files/ supporting files folder.

To proceed, go to Previewing the Menu in a Browser.

Previewing the Menu in a Browser

There are a few caveats to note when previewing your menu in a browser.

First, the menu can only be previewed from the HTML page. The SWF file cannot be viewed directly, as all necessary configuration data is passed in the HTML menu code.

Second, all of menu files (in the maximenu_files/ folder) must be uploaded prior to viewing on a remote server (e.g. http://www.myserver.com/somepage.htm).

Third, when using the Preview In Browser option in Dreamweaver, you may see a gray box instead of the menu. This indicates that you need to disable the following setting: Edit > Preferences... (Dreamweaver > Preferences... on Mac) > Preview In Browser > Preview using Temporary File.

Addressing Flash Player Security Warnings (sometimes evidenced by no menu appearing)

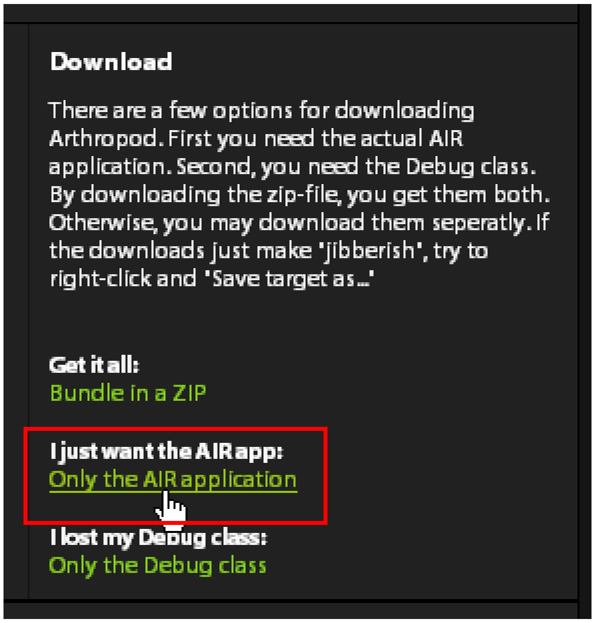
Fourth, external files including skin files and images may not be displayed when previewing your page from a local URL (e.g. file:///C:/mysite/somepage.htm or file:///Users/username/Documents/Sites/mysite/somepage.html). This applies ONLY to local URL's, prior to uploading your page to a remote server. Once the file has been uploaded to a remote server (or any server which is accessed via http://), you will no longer experience this Flash Player security restriction. To work around this issue when previewing locally, you must add the maximenu_files folder to your Flash Player trusted content. To do this, go to http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager04.html . Click the Edit Locations... button, and select Add Location. Click 'Browse for Folder', and select the maximenu_files/ supporting files folder. Click 'Confirm' to add this folder to your trusted content list. After doing so, reload your menu page.

Fifth, on Mac you may experience issue #4 even after updating your Flash Player security settings. This is due to the way Dreamweaver by default creates the preview file:/// URL. Dreamweaver will create a URL which includes the drive name, for example file:///**Macintosh HD/**Users/username/Documents/Sites/mysite/somepage.html. However, the Flash Player settings dialog usually trusts content based on a URL which has no drive name. Thus, you must remove the drive name in your browser's address bar. For example, the URL above should be changed to: file:///Users/username/Documents/Sites/mysite/somepage.html.

Problems #4 and #5 can be also evidenced by submenus failing to open when you hover over the menu. (The menu will not ever expand to its expanded size).

Debugging a Menu Preview

The menu file has been designed to display relevant preview error messages in an Adobe AIR based debugger named Arthropod. To run Arthropod, you must already have the Adobe AIR runtime installed (see the Installation Guide section of this manual). To install Arthropod, go to: http://arthropod.stopp.se/ . Click the 'I just want the AIR app' link to download and install the program:



Download

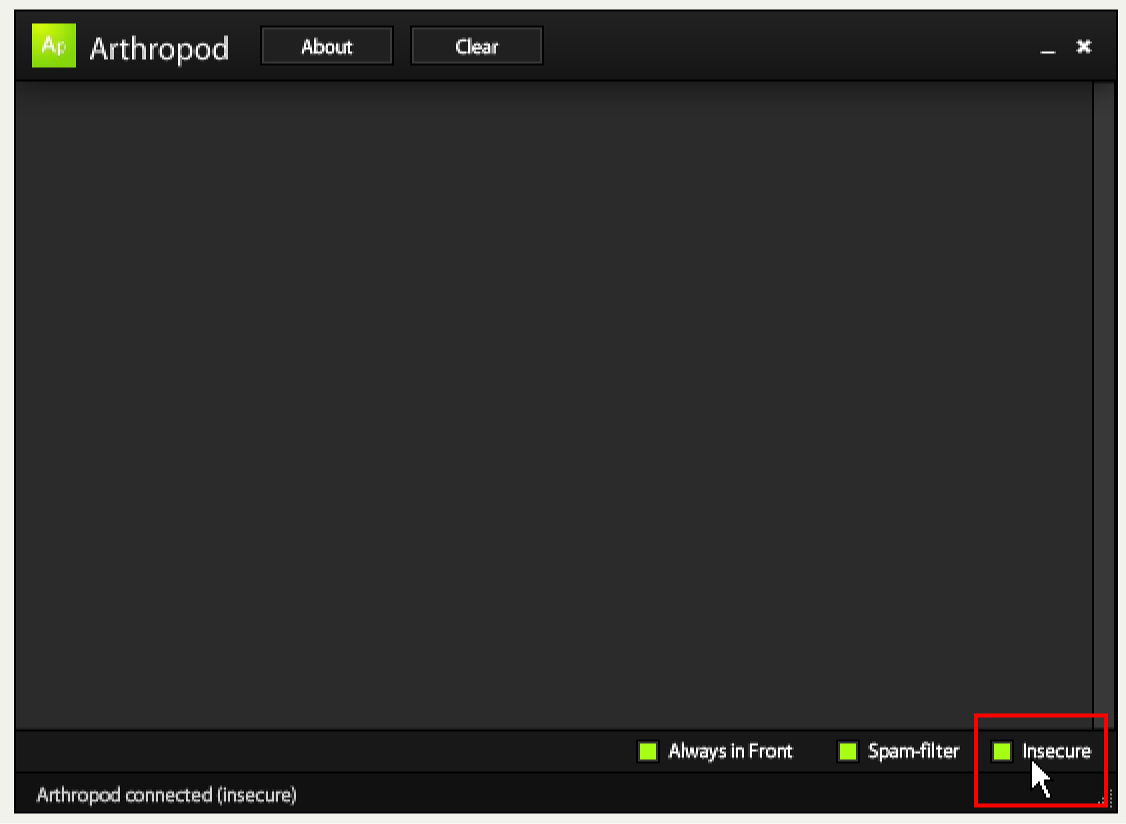
There are a few options for downloading Arthropod. First you need the actual AIR application. Second, you need the Debug class. By downloading the zip-file, you get them both. Otherwise, you may download them seperatly. If the downloads just make 'jibberish', try to right-click and 'Save target as...'

Get it all:
Bundle in a ZIP

I just want the AIR app:
Only the AIR application

I lost my Debug class:
Only the Debug class

Once you have the application installed, launch Arthropod. You should see a window like this:



Arthropod connected (insecure)

Prior to launching your menu preview to debug, **you must check the 'Insecure' button in the Arthropod window** (highlighted above). It is recommended you only enable this option while debugging your menu. Once the Arthropod window is running, you can switch to the browser and (re)load the preview page. You should see relevant error messages if any errors or problems occur. All MaxiMenu related messages will start with *'MaxiMenu: '*.

Using Dynamic Menu Data

Using External Data Files

Usually, you will pass menu data to a menu through HTML content embedded directly in your page. However, in certain situations, you may want to store all menu data in an external file. There are two main advantages to doing this:

- You can use dynamic data in the external file (PHP, ASP, Coldfusion, .NET, etc)
- All menus' data can be changed by only modifying one file. You can have only one data file for an entire site, rather than passing the same data to every page.
- The removal of unnecessary duplicated code makes your pages and menus load faster.

To utilize an external data file, a `content_url` parameter is passed to the menu. This parameter is the path to an external XML file containing a Content XML string. (See example 1 below.) When you specify an external data file, all content data passed directly in the page's HTML code will be ignored.

Example 1:

```
...
MaxiMenu({holder_id:'dwuser_maximenu_0',expanded_height:'500',normal_width:'600',swf:'maximenu_files/maximenu.swf',normal_height:'40',content_url:'maximenu_files/content_file.xml'});
...
```

You can store the external data file anywhere, but for organization it is recommended you place it in the maximenu_files/ directory. These files can be either static XML files, or dynamic ASP, PHP, Coldfusion, .NET, Java etc. files.

Content Format in External Data Files

You can use one of two different formats for external data files:

- HTML list format (this is the format used by the menu usually)
- XML format

HTML List Format Example:

```
<?xml version="1.0" encoding="utf-8"?> <ul> <li><a href="javascript:void(0)">Developer Tools</a> <ul> <li><a href="">Visual C#</a></li> <li><a href="">Visual Basic</a></li> <li><a href="">Visual C++</a></li> </ul> </li> <li><a href="">Platforms</a> <ul> <li><a href="">ASP.NET</a></li> <li><a href="">Internet Explorer</a> <ul> <li><a href="">Version 6</a></li> <li><a href="">Version 7</a></li> <li><a href="">Version 8 Beta</a></li> </ul> </li> <li><a href="">Flash Player</a></li> </ul> </li> <li><a href="">Architecture</a> <ul> <li><a href="">SQL Server</a></li> <li><a href="">.NET Framework</a></li> <li><a href="">Windows Server</a></li> </ul> </li> <li><a class="search" href="javascript:void(alert('You searched for \'{q}\'. In a real case, you would be sent to a search page.'));">Search:</a></li> </ul>
```

XML Format Example:

```
<?xml version="1.0" encoding="utf-8"?> <menu> <item label="Bla - 2" link=""
icon="MagnifyingGlass" iconOn="GlassOn"> <spacer label="Division 1" /> <item label="Bla -
2.1 Google " link="http://www.google.com/" /> <item label="Bla - 2.2" link=""> <item
label="Bla - 2.2.1" link="" /> <item label="Bla - 2.2.2" link="" /> <item label="Bla -
2.2.3" link="" /> </item> <spacer label="Division 2" /> <item label="Bla - 2.3" link="" >
<item label="Bla - 2.3.1" link="" /> <item label="Bla - 2.3.2" link="" /> <item
label="Bla - 2.3.3" link=""> <item label="Bla - 2.3.3.1" link="" /> <item label="Bla -
2.3.3.2" link="" /> <item label="Bla - 2.3.3.3" link="" /> </item> <item label="Bla -
2.3.4" link=""> <item label="Bla - 2.3.3.1" link="" /> <item label="Bla - 2.3.3.2"
link="" /> <item label="Bla - 2.3.3.3" link="" /> </item> </item> <item label="Bla - 2.4"
link="" /> </item> <item label="Bla - 3" link=""> <item label="Bla - 2.1 for" link="" />
<item label="Bla - 2.2" link=""> <item label="Bla - 2.2.1" link="" /> <item label="Bla -
2.2.2" link="" /> <item label="Bla - 2.2.3" link="" /> </item> <item label="Bla - 2.3"
link="" > <item label="Bla - 2.3.1" link="" /> <item label="Bla - 2.3.2" link="" /> <item
label="Bla - 2.3.3 on and on etc" link=""> <item label="Bla - 2.3.3.1" link="" /> <item
label="Bla - 2.3.3.2" link="" /> <item label="Bla - 2.3.3.3" link="" /> </item> <item
label="Bla - 2.3.4" link=""> <item label="Bla - 2.3.3.1" link="" /> <item label="Bla -
2.3.3.2" link="" /> <item label="Bla - 2.3.3.3" link="" > <item label="Bla - 2.3.3.3.1"
link=""></item> </item> </item> </item> <item label="Bla - 2.4" link="" /> </item>
<search label="Search:" link="http://www.google.com/search?q=\{q\}"
icon="MagnifyingGlass" /> <item label="Bla - 4 Contact" link="" /> </menu>
```

Either of these formats can utilize dynamic data.

Style XML Overview

All styles for a menu are controlled by the Styles XML passed to the menu. These can be edited most easily using the MaxiMenu Wizard. However, these values can also be edited manually. This section of the usage manual details how to manually edit the Style XML.

There are three methods by which the Style XML can be passed to a menu:

- Style XML string in page HTML - Passed through the menu HTML code, as the `styles` parameter. The XML is passed in a string value. (See example 1 below.)
- External Style XML file - In the menu HTML, a `styles_url` parameter is passed to the menu. This parameter is the path to an external XML file containing a Styles XML string. This allows for less code duplication, making it possible to only have one styles file for an entire site. (See example 2 below.)
- External skin file - In the menu HTML, a `skin_url` parameter is passed to the menu. This parameter is the path to an external skin file, which contains embedded Styles XML (see the Creating Skin Files guide). This allows for a central package for all icons, background images, styles, etc. (See example 3 below.)

Example 1:

```
...
MaxiMenu({holder_id:'dwuser_maximenu_0',expanded_height:'500',normal_width:'600',swf:'max
imenu_files/maximenu.swf',normal_height:'40',styles:'<styles>...</styles>'});
...
```

Example 2:

```
...
MaxiMenu({holder_id:'dwuser_maximenu_0',expanded_height:'500',normal_width:'600',swf:'max
imenu_files/maximenu.swf',normal_height:'40',styles_url:'maximenu_files/style_file.xml'})
;
...
```

Example 3:

```
...
MaxiMenu({holder_id:'dwuser_maximenu_0',expanded_height:'500',normal_width:'600',swf:'max
imenu_files/maximenu.swf',normal_height:'40',skin_url:'maximenu_files/custom_skin.swf'});
...
```

Style XML Format

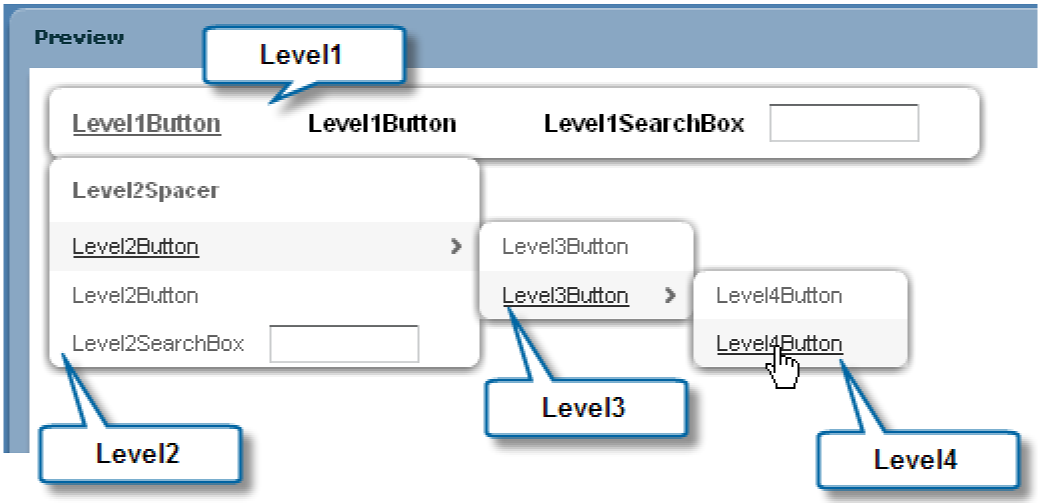
The Style XML has the following format:

```
<styles>
      <!-- *** Level 1 Styles *** -->
      <Level1Button ... />
      <Level1ButtonOn ... />
      <Level1Spacer ... />
      <Level1SearchBox ... />

      <!-- *** Level 2 Styles *** -->
      <Level2Button ... />
      <Level2ButtonOn ... />
      <Level2Spacer ... />
      <Level2SearchBox ... />

     <!-- *** Generic styles *** -->
     <SearchBox ... />
     <SubMenuArrow ... />
</styles>
```

There is a containing <styles> node which surrounds all other code. This node has several children, one to represent each item type on each level. Below is a schematic detailing how the level styles apply:



Based on the prefix attached to each item, it will control the associated level. For example, the <Level4ButtonOn ... /> node will control the fourth level's buttons on rollover.

The item styles apply as follows:

- **Level*N*Button** - Controls standard link items for the specified menu level. Also controls all menu-specific (not button-specific) styles; these are the styles which control the background, layout, etc for the entire level.
- **Level*N*ButtonOn** - Controls the hover-state styles for standard link items for the specified menu level
- **Level*N*Spacer** - Controls the styles for dividers on the specified menu level
- **Level*N*SearchBox** - Controls the styles for search boxes on the specified menu level

The Generic Styles control the following aspects:

- **SearchBox** - Controls generic styles for the search box items
- **SubMenuArrow** - Controls the styles for the arrow optionally displayed next to an item if it has a submenu. (In the schematic above, these are the arrows to the right of the Level2Button and Level3Button items.)

Style Inheritance

Level-based menu styles function much like cascading style sheets (CSS), inheriting most styles defined for their parent level. For example, the following inheritance chains exist:

Level3ButtonOn < Level2ButtonOn < Level1ButtonOn < Level1Button
Level2SearchBox < Level2Button < Level1Button
Level3Divider < Level3Button < Level2Button < Level1Button

Due to inheritance, most menus only need to have styles specifically defined for the Level1 and Level2 items. This will provide unique styles for the main menu level and all submenus respectively.

Available Node Attribute Details

The Style XML available node attribute documentation is broken into two main pages:

- Level Based Styles - Level*N*Button, Level*N*ButtonOn, Level*N*Spacer and Level*N*SearchBox
- Generic Styles - SearchBox and SubMenuArrow

All level-based styles are grouped together because of the common attributes shared by the elements. The only non-shared attributes are:

- The menu-specific styles on the Level*N*Button nodes. These attributes are irrelevant to the Level*N*ButtonOn, Level*N*Spacer and Level*N*SearchBox elements.

- The hover-specific styles on the Level*N*ButtonOn nodes only apply to the hover state of the corresponding LevelNButton nodes; the Spacer and SearchBox elements have no hover state, and are therefore unaffected by hover-specific styles.

Level Based Styles

The Level Based Styles details the available attributes for the Level*N*Button, Level*N*ButtonOn, Level*N*Spacer and Level*N*SearchBox nodes.

All level-based styles are grouped together because of the common attributes shared by the elements. The only non-shared attributes are:

- The menu-specific styles on the Level*N*Button nodes. These attributes are irrelevant to the Level*N*ButtonOn, Level*N*Spacer and Level*N*SearchBox elements.
- The hover-specific styles on the Level*N*ButtonOn nodes only apply to the hover state of the corresponding LevelNButton nodes; the Spacer and SearchBox elements have no hover state, and are therefore unaffected by hover-specific styles.

Notes:

- When images are specified, the values represent either the URL to the image, or the name of the image embedded in the skin file.
- When opacities are specified, 0.00 represents 0% opaque / 100% transparent, and 1.00 represents 100% opaque / 0% transparent.

Attribute	Applies to	Description	Possible Values
		=== Menu Background and Layout ===	
layoutDirection	Level*N*Button	Specifies whether the menu is vertically or horizontally displayed	horizontal, vertical
drawExtraBackground	Level*N*Button	Only applicable if you specify dimensions for the menu (using menuWidth or menuHeight). If enabled and the content doesn't fill the available width, then the menu background will be drawn to match the dimensions specified, regardless of content size.	true, false
menuWidth	Level*N*Button	Only applicable if you choose to use the Draw Extra Background feature. Specifies the width of the menu. A value of '0' is equal to auto (the contents' dimensions will be used).	*Number*
menuHeight	Level*N*Button	Only applicable if you choose to use the Draw Extra Background feature with a vertical layout. Specifies the height of the menu. A value of '0' is equal to auto (the contents' dimensions will be used).	*Number*
childPosition	Level*N*Button	Specifies whether any child	right, bottom,

Attribute	Applies to	Description	Possible Values
		menus will be displayed to the right or the bottom of the hovered button. If you select 'Bottom + Justified', the submenu will be aligned horizontally with the left edge of its parent menu. If you select 'Right + Justified', the submenu will be aligned vertically with the top of its parent menu.	rightJustified, bottomJustified
menuBackgroundColor	Level*N*Button	Specifies the menu background color, displayed behind the entire menu level.	*Hexadecimal Color*
menuBackgroundAlpha	Level*N*Button	Specifies the opacity of the menu background color, displayed behind the entire menu level.	*Numerical, 0.00 to 1.00*
menuBorderColor	Level*N*Button	Specifies the color of the border around the entire menu level.	*Hexadecimal Color*
menuBorderWidth	Level*N*Button	Specifies the width of the border around the entire menu level. Negative one (-1) makes no border; zero (0) makes a hairline border.	*Number*
menuCornerRadius	Level*N*Button	Specifies the radius of the corner curves around the entire menu level.	*Number*
menuShadow	Level*N*Button	Specifies whether a drop shadow is shown behind the entire menu level.	true, false
menuShadowColor	Level*N*Button	(Not in wizard) Specifies the color of the drop shadow shown behind the entire menu level (if shadow is enabled).	*Hexadecimal Color*
menuShadowSize	Level*N*Button	Specifies the width of the drop shadow shown behind the entire menu level (if shadow is enabled).	*Number*
menuImage	Level*N*Button	Specifies the image shown behind the entire menu level. Specify 'null' to use no image.	*Image URL or embed ID*
menuImageScaleGrid	Level*N*Button	Specifies the scale-9 slicing grid used for the image shown behind the entire menu level. The format is: top,right,bottom,left. Do not enter any extra spaces between the commas and numbers.	n,n,n,n *where n is a number*
menuImagePositionGrid	Level*N*Button	Specifies the positioning offsets for the image shown behind the entire menu level. The format is: top,right,bottom,left. Do not enter any extra spaces between the commas and numbers.	n,n,n,n *where n is a number*
menuEffect	Level*N*Button	Only applies to submenus (after first level). Specifies the type of transition effect used to hide and show the menu.	*Fade:* fade, *Zoom + Fade (Large):* zoomFadeBig, *Zoom + Fade (Small):* zoomFadeSmall*, Drop Down:* dropDown*, Drop Down + Fade:* dropDownFade*, Drop Left:* dropLeft*, Drop Left*

Attribute	Applies to	Description	Possible Values
			+ Fade: dropLeftFade*, Drop Diagonal:* dropDiagonal*, Drop Diagonal + Fade:* dropDiagonalFade*, Blur:* blur*, Blur + Fade:* blurFade*, Wipe Left:* wipeLeft*, Wipe Top:* wipeTop*, (None):* none *Value list:* fade, zoomFadeBig, zoomFadeSmall, dropDown, dropDownFade, dropLeft, dropLeftFade, dropDiagonal, dropDiagonalFade, blur, blurFade, wipeLeft, wipeTop, none
menuEffectSpeed	Level*N*Button	Only applies to submenus (after first level). Specifies the duration, in milliseconds, of the transition effect used to hide and show the menu.	*Number*
		=== Button Background and Layout ===	
useArrow	Level*N*Button	Specifies whether each button with a submenu displays an arrow indicating this.	true, false
image	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the image shown behind each button. Specify 'null' to use no image.	*Image URL or embed ID*
imageScaleGrid	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the scale-9 slicing grid used for the image shown behind each button. The format is: top,right,bottom,left. Do not enter any extra spaces between the commas and numbers.	n,n,n,n *where n is a number*
imagePositionGrid	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the positioning offsets for the image shown behind each button. The format is: top,right,bottom,left. Do not enter any extra spaces between the commas and numbers.	n,n,n,n *where n is a number*
imageOn	Level*N*Button	Specifies the hover-state image shown behind each button. Specify 'null' to use no image.	*Image URL or embed ID*
imageScaleGridOn	Level*N*Button	Specifies the scale-9 slicing grid used for the hover-state image shown behind each button. The format is: top,right,bottom,left. Do not enter any extra spaces between the commas and numbers.	n,n,n,n *where n is a number*

Attribute	Applies to	Description	Possible Values
imagePositionGridOn	Level*N*Button	Specifies the positioning offsets for the hover-state image shown behind each button. The format is: top,right,bottom,left. Do not enter any extra spaces between the commas and numbers.	n,n,n,n *where n is a number*
vPadding	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the padding between the top and bottom of the label text and the top and bottom of each button.	*Number*
hPadding	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the padding between the left edge of the label text and the left edge of each button.	*Number*
hPaddingRight	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the padding between the right edge of the label text and the right edge of each button.	*Number*
backgroundColor	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies each button's background color.	*Hexadecimal Color*
backgroundAlpha	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the opacity of each button's background color.	*Numerical, 0.00 to 1.00*
bevelStrength	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the opacity of the background fade / bevel on each button's background color.	*Number, 0 to 100*
borderColor	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the color of the border around each button.	*Hexadecimal Color*
borderWidth	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the width of the border around each button. Negative one (-1) makes no border; zero (0) makes a hairline border.	*Number*
cornerRadius	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the radius of the corner curves around each button.	*Number*
		=== Button Text Styles ===	
font	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the font displayed on each button. The *embedFonts* attribute specifies whether this is an embedded font. See the *embedFonts* attribute to view more information about the embed option.	*Font name or embed ID*
embedFonts	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	This option should only be set to *true* if you are using a font which is embedded in a skin file or loaded from an external font file. Enabling this option when it should not be enabled can result in text not appearing. To load fonts from an external file, create a new	true, false

Attribute	Applies to	Description	Possible Values
		FLA file and place desired font instances in the library. Enable 'Export for Actionscript', and specify 'MenuFont' as the linkage ID. For subsequent fonts, specify MenuFont1, MenuFont2, etc. Export the FLA to a Flash Player 9 SWF file which is located at *fonts/menufonts.swf* relative to the menu's holding HTML page.	
size	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the size, in points, of the label text displayed on each button.	*Number*
textColor	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the color of the label text on each button.	*Hexadecimal Color*
bold	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies whether the label text on each button is bold or not.	true, false
italic	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies whether the label text on each button is italic or not.	true, false
underline	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies whether the label text on each button is underlined or not.	true, false
textAlign	Level*N*Button, Level*N*Spacer, Level*N*SearchBox	Specifies the alignment for the label text on each button. Generally, 'left' is the choice to always be used.	left, center, right
		=== Button Hover Effects (Only Applies to Button Styles) ===	
		Note: For the hover-state button background image, see the Button Background section above.	
leaveBackgroundText	Level*N*ButtonOn	(Not in wizard.) Specifies whether or not the up-state text will remain after the hover text has been revealed. It is best to leave this as true to prevent occasional blinking text.	true, false
rolloverDelay	Level*N*ButtonOn	(Not in wizard.) Specifies the delay, in milliseconds, before a button registers a rollover event, switches to the hover-state, and opens any submenus. The default value is 10. Specifying a greater value will make the menu less responsive, but will also make it more difficult to lose ones place and inadvertently open another submenu.	*Number*

Attribute	Applies to	Description	Possible Values
effectSpeed	Level*N*ButtonOn	Specifies the duration, in milliseconds, of the transition from the standard state to the hover/selected state.	*Number*
backgroundColor	Level*N*ButtonOn	Specifies each button's background color when the button is selected.	*Hexadecimal Color*
backgroundAlpha	Level*N*ButtonOn	Specifies the opacity of each button's background color when the button is selected.	*Numerical, 0.00 to 1.00*
textColor	Level*N*ButtonOn	Specifies the color of the label text on each button when the button is selected.	*Hexadecimal Color*
bold	Level*N*ButtonOn	Specifies whether the label text on each button is bold or not when the button is selected. This option is not recommended, because it will distort the text size (and button layout) if not using a monospace font.	true, false
italic	Level*N*ButtonOn	Specifies whether the label text on each button is italic or not when the button is selected.	true, false
underline	Level*N*ButtonOn	Specifies whether the label text on each button is underlined or not when the button is selected.	true, false
		=== Generic Styles ===	
image	SubMenuArrow	Specifies the icon image shown for the arrow. If an icon image is used, the default line-based arrow will not be drawn. Specify 'null' to use no image.	*Image URL or embed ID*
imageOn	SubMenuArrow	Specifies the hover-state icon image shown for the arrow. Only applies if a default arrow image has been applied. Specify 'null' to use no image.	*Image URL or embed ID*
lineColor	SubMenuArrow	Specifies the color of the lines which comprise the default line-based arrow.	*Hexadecimal Color*
lineWidth	SubMenuArrow	Specifies the width of the lines which comprise the default line-based arrow. Negative one (-1) makes no line; zero (0) makes a hairline line.	*Number*
size	SubMenuArrow	Specifies the size, in pixel width, of the default line-based arrow.	*Number*
boxWidth	SearchBox	Specifies the width of the text field on the search box(es).	*Number*
boxHeight	SearchBox	Specifies the height of the text field on the search box(es).	*Number*

Documentation for Generic Styles.

Back to Style XML Overview.

Generic Styles

The Generic Styles table below details the attributes available for the non-level-specific SearchBox and SubMenuArrow nodes.

Attribute	Applies to	Description	Possible Values
		=== SubMenuArrow ===	
`image`	`SubMenuArrow`	Specifies the icon image shown for the arrow. If an icon image is used, the default line-based arrow will not be drawn. Specify 'null' to use no image.	*Image URL or embed ID*
`imageOn`	`SubMenuArrow`	Specifies the hover-state icon image shown for the arrow. Only applies if a default arrow image has been applied. Specify 'null' to use no image.	*Image URL or embed ID*
`lineColor`	`SubMenuArrow`	Specifies the color of the lines which comprise the default line-based arrow.	*Hexadecimal Color*
`lineWidth`	`SubMenuArrow`	Specifies the width of the lines which comprise the default line-based arrow. Negative one (-1) makes no line; zero (0) makes a hairline line.	*Number*
`size`	`SubMenuArrow`	Specifies the size, in pixel width, of the default line-based arrow.	*Number*
		=== SearchBox ===	
`boxWidth`	`SearchBox`	Specifies the width of the text field on the search box(es).	*Number*
`boxHeight`	`SearchBox`	Specifies the height of the text field on the search box(es).	*Number*
`textColor`	`SearchBox`	Specifies the color of the text in the text field on the search box(es).	*Hexadecimal Color*

Documentation for Level Based Styles.

Back to Style XML Overview.

Creating Skin Files

Skins are special SWF files which act as a package to hold all styles and style assets for a menu, including Style XML, images/icons, and embedded fonts. This section of the manual details the process of creating a skin file.

Prerequisites:

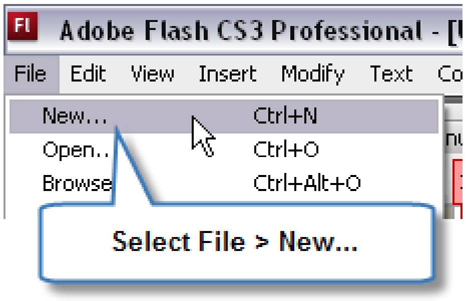
- Adobe Flash CS3 or higher
- Some experience using Flash and editing Actionscript code

Sections in this guide:

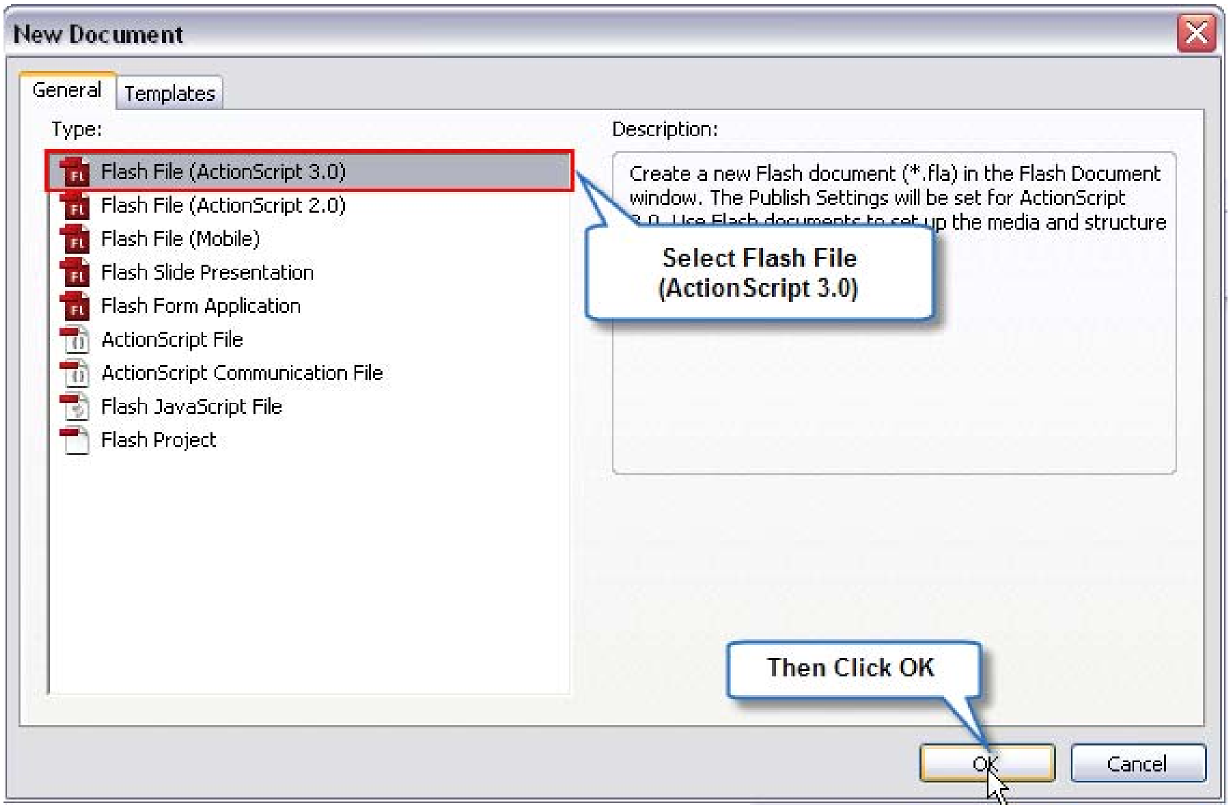
- Creating the FLA file
- Adding Styles XML
- Embedding Images / Icons
- Embedding Fonts
- Exporting the Skin File

Creating the FLA file

To create a SWF file, you must use Adobe Flash CS3 or higher (CS4, etc). In Flash, select File > New...:



In the New Document dialog, select Flash File (Actionscript 3.0), then click OK:



Create a new Flash document (*.fla) in the Flash Document window. The Publish Settings will be set for ActionScript 3.0. Use Flash documents to set up the media and structure

Select Flash File (ActionScript 3.0)

Then Click OK

Save the file in your (site root)/maximenu_files/ folder.

Adding Styles XML

Open the Actions panel for Frame 1 of the main timeline. In this location, you will specify a function which returns the Style XML to be used for the menu. This function must be named getStylesXML(). This function will receive one parameter when it is called, an object which contains any skin-specific variables passed through the menu HTML code (as indicated by a prefix of skin_...). The return value must be an XML object, formatted using the Style XML Format. Usually, you will just return a pre-defined XML variable, as shown in this example:

[Code Listing]

```
var menuStylesXML:XML =
<styles>
	<Level1Button childPosition="bottom" menuShadow="false" menuBackgroundAlpha="0.00"
menuImage="MenuImage0" menuImageScaleGrid="15,15,15,15" layoutDirection="horizontal"
menuWidth="0" menuHeight="300" useArrow="false" cornerRadius="0"
drawExtraBackground="false" borderWidth="-1" underline="false" backgroundColor="#FFFFFF"
backgroundAlpha="0.00" bevelStrength="00" hPadding="10" hPaddingRight="30" vPadding="8"
textColor="#000000" embedFonts="false" font="Arial" size="12" bold="true" />
	<Level1ButtonOn image="" effectSpeed="500" backgroundColor="#000000"
backgroundAlpha="0" textColor="#666666" underline="true" />

	<Level2Button menuImage="MenuImage1" menuImagePositionGrid="-5,-5,-5,-5"
menuShadow="false" menuBackgroundAlpha="0" image="null" layoutDirection="vertical"
menuCornerRadius="12" menuBackgroundColor="#FFFFFF" useArrow="true" underline="false"
backgroundColor="#CCCCCC" backgroundAlpha="0" borderWidth="-1" borderColor="#999999"
bevelStrength="0" hPadding="10" hPaddingRight="30" vPadding="3" textColor="#666666"
embedFonts="false" font="Arial" size="11" bold="false" />
```

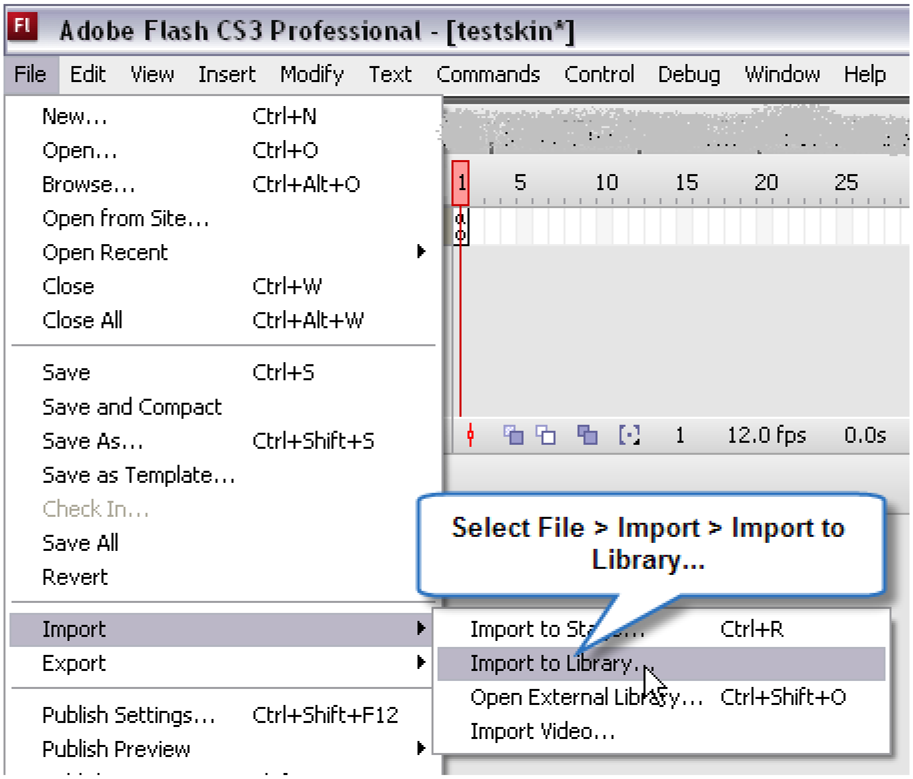
```
        <Level2Spacer useArrow="true" underline="false" backgroundColor="#CCCCCC"
backgroundAlpha="0" borderWidth="-1" borderColor="#999999" bevelStrength="0"
hPadding="10" hPaddingRight="30" vPadding="7" textColor="#666666" embedFonts="false"
font="Arial" size="11" italic="false" bold="true" />
        <Level2ButtonOn effectSpeed="500" underline="true" backgroundAlpha="0.50"
backgroundColor="#EDEDED" textColor="#333333" />
        <SubMenuArrow lineWidth="1" lineColor="#FF9900" size="6"
image="images/arrow_7.gif" imageOn="images/arrow_6.gif" />
        <SearchBox boxWidth="75" boxHeight="19" />

</styles>function getStylesXML(vars:Object):XML {
        if (vars.skin_themecolor != null)
        {
                menuStylesXML.Level1Button.@backgroundColor = vars.skin_themecolor;
        }
        return menuStylesXML;
}
```

In this example, the getStylesXML function checks to see if a skin_themecolor parameter has been passed. If it has, it updates the menuStylesXML XML variable to use the new background color. Finally, the menuStylesXML variable is returned.

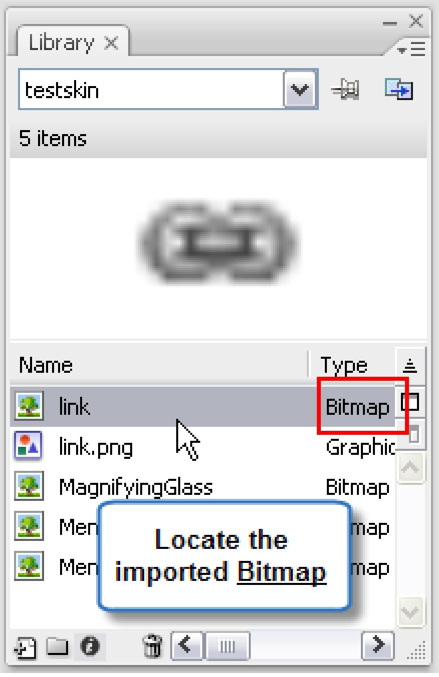
Embedding Images / Icons

If you wish to include background images or icons in your menu, these files can easily be embedded in the skin file. To do so, go to File > Import > Import to Library... in Flash:



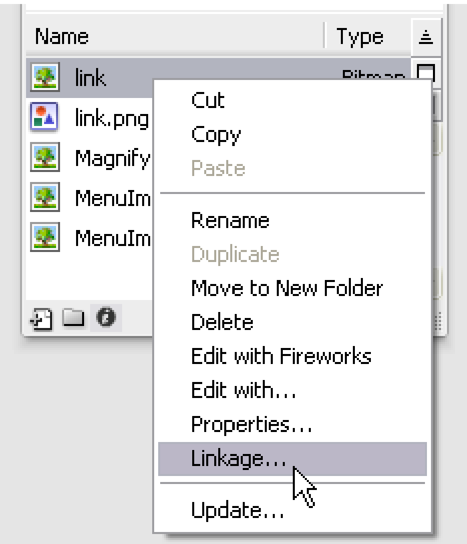
Select File > Import > Import to Library...

Select the image you wish to import and click OK. Next, open the Library panel, and locate the Bitmap instance you just imported:

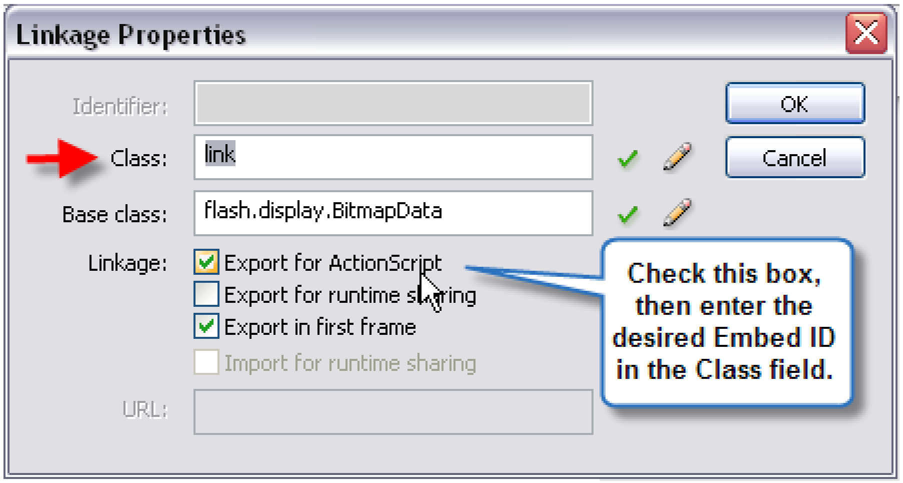


Locate the imported Bitmap

Right-click (Mac Cmd-click) the instance, and select Linkage...:



In the Linkage dialog, check the Export for Actionscript option, then enter the desired Embed ID in the Class field:



Identifier:

Class: link

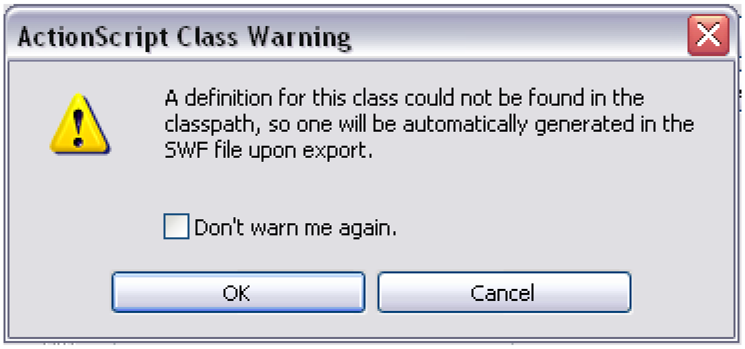
Base class: flash.display.BitmapData

Linkage:
☑ Export for ActionScript
☐ Export for runtime sharing
☑ Export in first frame
☐ Import for runtime sharing

URL:

Check this box, then enter the desired Embed ID in the Class field.

Click OK. If you receive the following warning message, click OK:



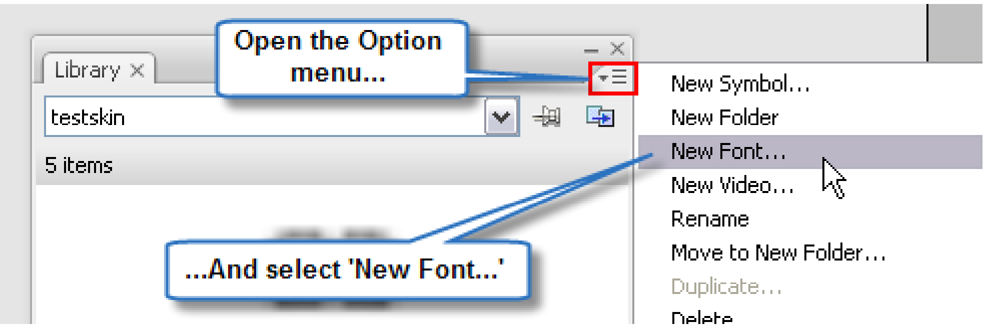
A definition for this class could not be found in the classpath, so one will be automatically generated in the SWF file upon export.

☐ Don't warn me again.

On export, your newly embedded image will be able to be referenced by specifying **link** (or whatever you entered in the Class field) as the image source / embed ID.

Embedding Fonts

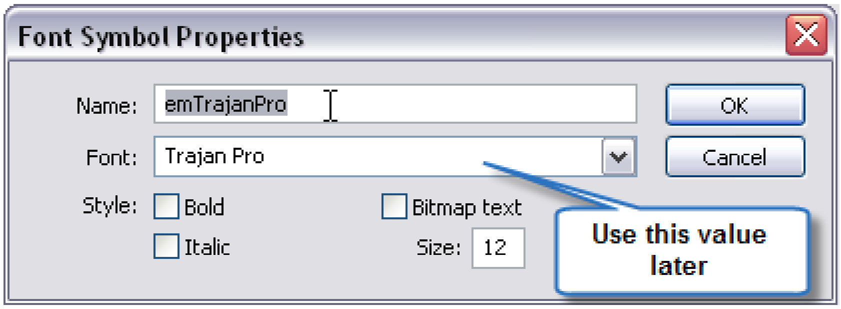
If you wish to use custom embedded fonts in your menu, these files can easily be included in the skin file. To do so, open the Library panel in Flash and select the Option menu on the right. Select the New Font... option:



Open the Option menu...

...And select 'New Font...'

In the Font Symbol Properties window which opens, select the font you want to embed and enter a unique name as the Font Name. The value in the Font field is the value you will later use to reference the font in the Styles XML.


Font Symbol Properties

Name: emTrajanPro

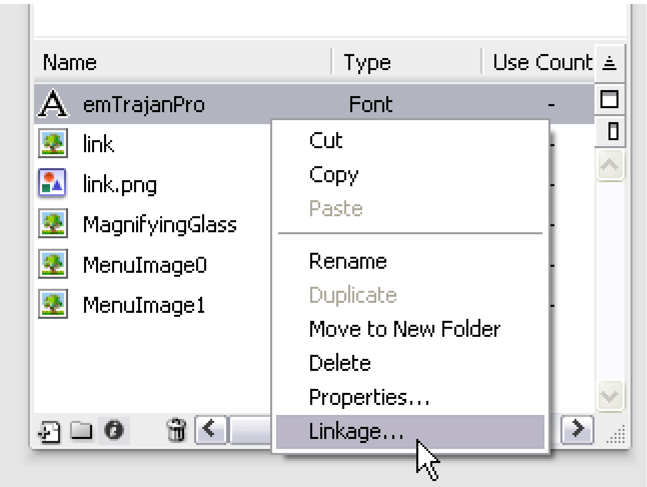
Font: Trajan Pro

Style: ☐ Bold ☐ Italic

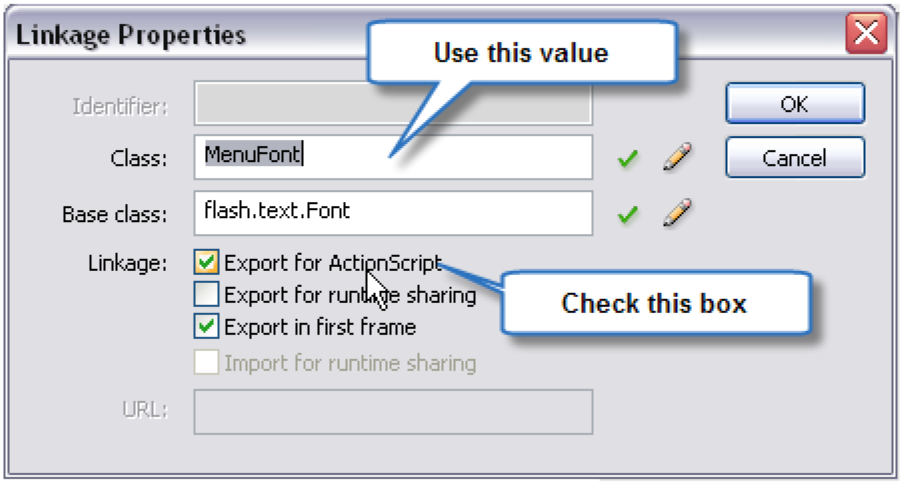
☐ Bitmap text

Size: 12

Be aware that checking the 'Bitmap text' option may cause the font to fail to display and is therefore not recommended. When you are finished, click OK. The new font symbol will appear in the Library panel. Right-click (Mac Cmd-click) the font and select Linkage...:


Name	Type	Use Count
emTrajanPro	Font	-
link		-
link.png		-
MagnifyingGlass		-
MenuImage0		-
MenuImage1		-

In the Linkage dialog, check the Export for Actionscript option. Next, you must enter **MenuFont** as the Class value. For subsequent fonts, use **MenuFont1**, **MenuFont2**, etc. Then click OK:


Linkage Properties

Identifier:

Class: MenuFont

Base class: flash.text.Font

Linkage: ☑ Export for ActionScript

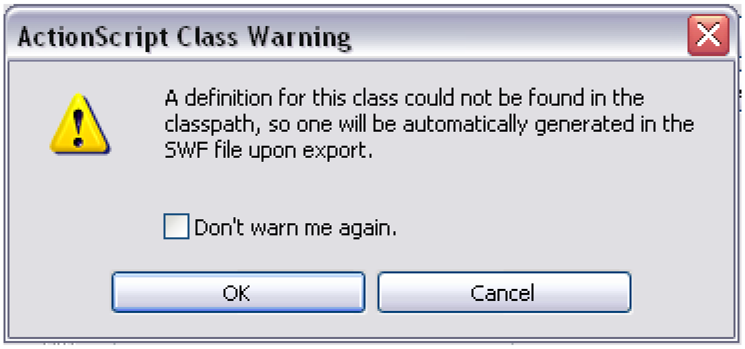
☐ Export for runtime sharing

☑ Export in first frame

☐ Import for runtime sharing

URL:

If you receive the following warning message, click OK:


ActionScript Class Warning

A definition for this class could not be found in the classpath, so one will be automatically generated in the SWF file upon export.

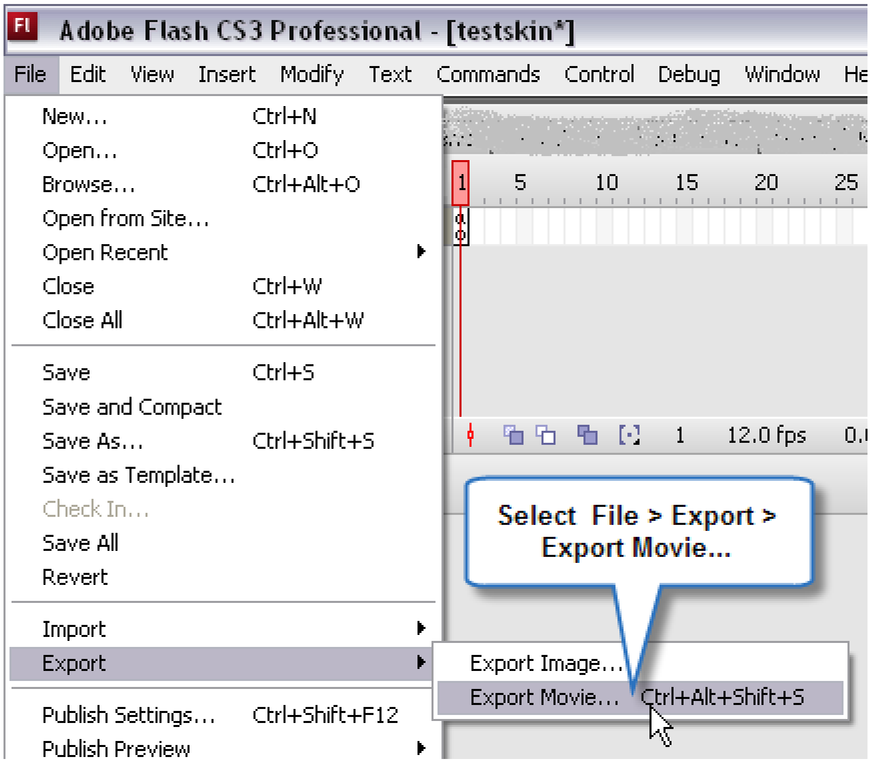
☐ Don't warn me again.

Your font is now embedded and you can export the skin file. Later, you will reference the name you selected in the 'Font' field in the Font Symbol Properties dialog as the font name in the Styles XML. Make sure that you set the embedFonts Style XML attribute to true; otherwise, the embedded font will not be displayed. In the Maximenu Wizard, this option is represented by the 'Embedded Font' checkbox directly below the font name.

Important Note: When previewing a menu inside the development wizard, all fonts must be installed locally on your system to preview properly. If you are using a skin file which has embedded fonts which are not on your system, the preview may display the default Times New Roman font. This behavior is only present when previewing a menu; it will not appear when viewing the menu in a browser.

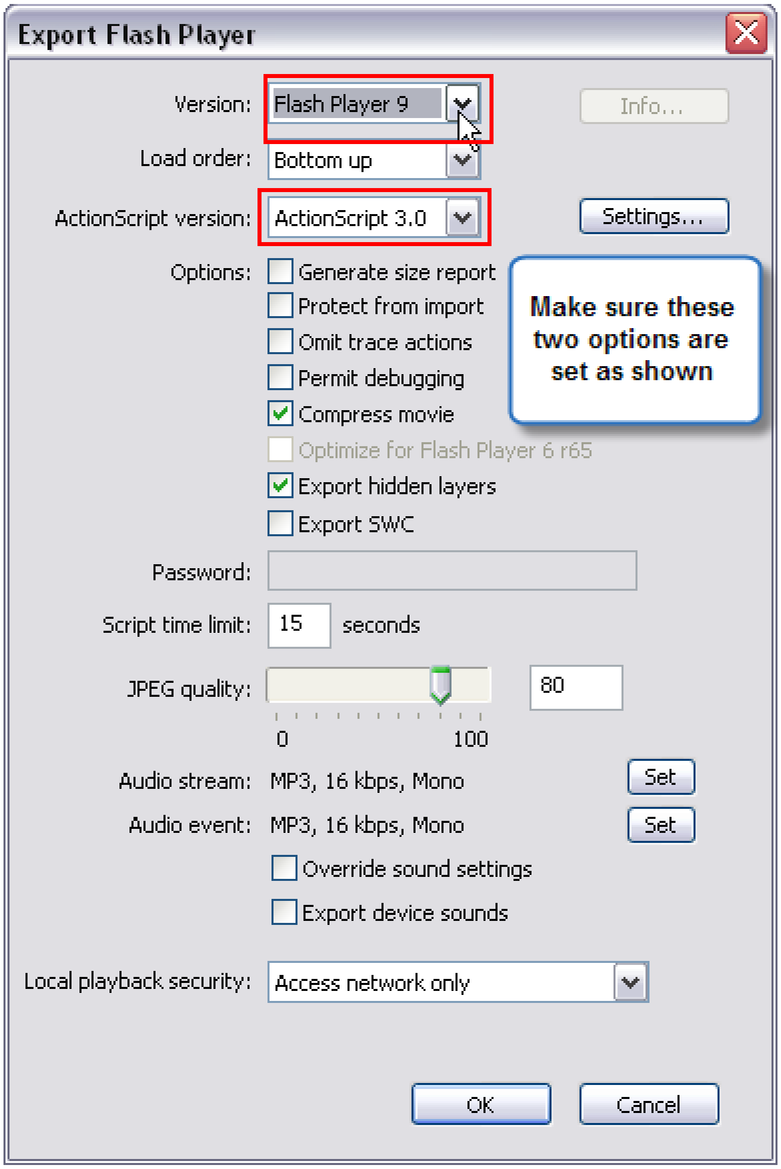
Exporting the Skin File

Once you have finished creating the skin file, select File > Export > Export Movie... to export the skin as a SWF file:



Select File > Export > Export Movie...

Specify the desired location - the (site root)/maximenu_files/ folder is recommended -, enter a file name, and click Save. In the Export Flash Player dialog, make sure that the Version option is set to Flash Player 9, and the ActionScript version is set to ActionScript 3.0:



Export Flash Player

Version: Flash Player 9

Load order: Bottom up

ActionScript version: ActionScript 3.0

Options:
☐ Generate size report
☐ Protect from import
☐ Omit trace actions
☐ Permit debugging
☑ Compress movie
☐ Optimize for Flash Player 6 r65
☑ Export hidden layers
☐ Export SWC

Password:

Script time limit: 15 seconds

JPEG quality: 80

Audio stream: MP3, 16 kbps, Mono

Audio event: MP3, 16 kbps, Mono

☐ Override sound settings
☐ Export device sounds

Local playback security: Access network only

Make sure these two options are set as shown

Your skin file is now packaged and ready to use or share. To apply the skin file to a menu, use the Menu Skins section of the MaxiMenu Wizard. Browse to the skin location and click (Re)Apply Skin to apply the skin.